

Редакционная коллегия

Баракина Т. В.
Бешенков С. А.
Босова Л. Л.
Воеводин Вл. В.
Дергачева Л. М.
Заславская О. Ю.
Захарова Т. Б.
Зенкина С. В.
Кириченко И. Б.
Кравцова А. Ю.
Кузнецов А. А.
Лаптев В. В.
Левченко И. В.
Окулов С. М.
Рыбаков Д. С.
Слинкина И. Н.

Редакция

Босова Л. Л.
главный редактор
Кириченко И. Б.
*заместитель
главного редактора*
Губкин В. А.
Коптева С. А.
Кузнецова Е. А.
Меркулова Н. И.
Федотов Д. В.
Шарапова Л. М.

Адрес редакции:

119261, г. Москва,
Ленинский пр-т, д. 82/2, комн. 6

Почтовый адрес:

119270, г. Москва, а/я 15

Телефон/факс: (495) 140-19-86

E-mail: readinfo@infojournal.ru

URL: <http://www.infojournal.ru>

Подписные индексы

в каталоге «Роспечать»:

для индивидуальных подписчиков — 81407
для предприятий и организаций — 81408

Подписано в печать 23.10.2017.

Формат 60×90¹/₈. Усл. печ. л. 8,0.

Тираж 2000 экз. Заказ № 235.

Отпечатано в типографии

ООО «Принт сервис групп»

105187, г. Москва, Борисовская ул.,
д. 14, стр. 6, тел./факс: (499) 785-05-18,
e-mail: 3565264@mail.ru

Свидетельство о регистрации

средства массовой информации

ПИ № 77-12068 от 11 марта 2002 г.

Все права защищены. Никакая часть журнала
не может быть воспроизведена в любой форме
или любыми средствами, электронными или
механическими, включая фотографирование,
сканирование, магнитную запись, размещение
в Интернете или иные средства копирования
или сохранения информации, без письменного
разрешения издательства.

Журнал входит в Перечень российских
рецензируемых научных изданий ВАК,
в которых должны быть опубликованы
основные научные результаты диссертаций
на соискание ученых степеней доктора
и кандидата наук

© «Образование и Информатика», 2017

ИНФОРМАТИКА В ШКОЛЕ

№ 8 (131) октябрь 2017

Содержание

От редакции 4

МЕТОДИКА ОБУЧЕНИЯ ИНФОРМАТИКЕ: ОПЫТ МПГУ

Кушниренко А. Г., Леонов А. Г. «КуМир» — учебное программное
обеспечение базового курса информатики..... 5

Босова Л. Л., Сивкова Е. О. Элементы теории множеств в школьном
курсе информатики 11

Самылкина Н. Н., Калинин И. А. Влияние образовательной
робототехники на содержание курса информатики основной школы 16

Салахова А. А. Техническое творчество и соревнования
для формирования новых качеств личности (на примере
робототехнических соревнований) 22

Бобров В. А. Методы решения задач на определение соответствий..... 25

Арнаутова Н. А. Применение самодельной мультипликации на разных
этапах урока 30

Воробьева В. А. Использование технологии опережающего обучения
при проведении практических работ по информатике в основной школе..... 33

Тарапата В. В. Лабораторная работа по основам микроэлектроники
«Асинхронный RS-триггер» в углубленном курсе информатики десятого
класса..... 36

Глотова М. Ю., Самохвалова Е. А. Использование PowerPoint
для визуализации информации..... 41

СУПЕРКОМПЬЮТЕРНОЕ ОБРАЗОВАНИЕ В ШКОЛЕ

Плаксин М. А. Пропедевтика параллельных вычислений в школьной
информатике. Многопоточность и многозадачность..... 48

МЕТОДИЧЕСКАЯ КОПИЛКА

Оленев А. А., Малиатаки В. В. Моделирование логических элементов
и простейших узлов ЭВМ в системе компьютерной математики Maple..... 58

XIV ВСЕРОССИЙСКИЙ КОНКУРС НАУЧНО-ПРАКТИЧЕСКИХ РАБОТ ПО МЕТОДИКЕ ОБУЧЕНИЯ ИНФОРМАТИКЕ И ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ ИНФО-2017

**Издательство «Образование и Информатика»,
Всероссийское научно-методическое общество педагогов
объявляют о проведении в 2017 году
конкурса по следующим номинациям:**

- 1. Программируем по-новому.**
- 2. Образовательные технологии для достижения метапредметных результатов.**
- 3. Правила информационной безопасности — изучаем и соблюдаем.**
- 4. Дистанционные технологии в практике работы образовательной организации.**
- 5. Инновации в профессиональной подготовке будущего учителя информатики.**

Оргкомитет конкурса

Руководит конкурсом **Организационный комитет** (далее — Оргкомитет), состоящий из представителей Российской академии образования, ведущих методистов, членов Всероссийского научно-методического общества педагогов, членов редакционных коллегий журналов «Информатика и образование» и «Информатика в школе», сотрудников объединенной редакции журналов.

Цели и задачи конкурса

1. Поддержка и распространение опыта педагогов и образовательных организаций по внедрению в образовательную практику современных методов и средств обучения и управления образованием.
2. Выявление и поддержка талантливых педагогов, методистов, руководителей образовательных организаций и органов управления образованием, заинтересованных в развитии инновационных образовательных технологий.
3. Включение педагогов, методистов, руководителей образовательных организаций и органов управления образованием в деятельность по разработке нового содержания образования, новых образовательных технологий, методик обучения и управления образованием.
4. Создание информационно-образовательного пространства на сайте издательства «Образование и Информатика», а также на страницах журналов «Информатика и образование» и «Информатика в школе» по обмену и распространению опыта внедрения инновационных образовательных технологий.
5. Повышение информационной культуры и информационно-коммуникационной компетентности всех участников образовательного процесса.

Условия участия в конкурсе

1. Участником конкурса может стать любой человек, связанный с работой в системе образования.
2. Возраст участников не ограничен.
3. Участником конкурса может быть индивидуальный заявитель или группа авторов.
4. Участниками конкурса могут быть как граждане России, так и граждане других стран, приславшие свои материалы на русском языке.
5. Участник конкурса может подать по одной заявке в каждой номинации.
6. Заявки на участие в конкурсе принимаются только через заполнение формы на сайте издательства «Образование и Информатика».
7. Форма участия в конкурсе — заочная.
8. **В дополнение к основному конкурсу** каждая работа может быть представлена автором для онлайн-голосования на сайте издательства «Образование и Информатика».
9. **Новое в конкурсе ИНФО! В дополнение к участию в конкурсе** каждый автор может представить тезисы своей работы для их публикации в специальном сборнике.

Сроки и этапы проведения конкурса

1. **Работы на конкурс принимаются** с 1 августа по 15 ноября 2017 года включительно. Работы, присланные позже 15 ноября 2017 года, к участию в конкурсе допускаться не будут.
2. **Голосование на сайте** за работы, представленные для онлайн-голосования, будет проходить с 1 по 20 декабря 2017 года включительно.
3. **Итоги конкурса** будут подведены до 31 января 2018 года и опубликованы на сайте издательства «Образование и Информатика», а также в журналах «Информатика и образование» и «Информатика в школе» № 1-2018.
4. **Лучшие работы** будут опубликованы в журналах «Информатика и образование» и «Информатика в школе».

Критерии оценки конкурсных работ

1. Оригинальность раскрытия темы, творческий потенциал, наличие самостоятельных идей, новизна и актуальность работы.
2. Использование инновационных педагогических технологий, разнообразие и целесообразность методических приемов.
3. Возможность масштабирования работы и проецирования на другие образовательные организации.
4. Системность и структурированность изложения материала.
5. Стилистически и орфографически грамотное изложение материала.
6. Наличие авторского дидактического обеспечения (мультимедийная презентация, видеоролик, интерактивный тест, сайт и т. д.).

Победители конкурса получают (бесплатно):

1. Диплом от Всероссийского научно-методического общества педагогов и издательства «Образование и Информатика».
2. Электронную подписку на журналы «Информатика и образование» и «Информатика в школе» на 2018 год.
3. По одному печатному экземпляру журналов «Информатика и образование» № 1-2018 и «Информатика в школе» № 1-2018, в которых будут опубликованы итоги конкурса.
4. Авторский печатный экземпляр журнала с опубликованной работой.

Победители онлайн-голосования будут отмечены **специальными дипломами**. Получение специального диплома по итогам онлайн-голосования не ограничивает получение участником диплома жюри в соответствующей номинации за ту же работу (то есть за одну и ту же работу участник может получить два диплома — специальный диплом по итогам онлайн-голосования и диплом жюри).

Сборник тезисов конкурса

По материалам конкурса будет издан сборник тезисов конкурсных работ. Сборник издается как **«Сборник научных трудов по материалам XIV Всероссийского конкурса научно-практических работ по методике обучения информатике и информатизации образования ИНФО-2017»**.

Публикация тезисов в сборнике — платная.

Подробная информация о стоимости публикации в сборнике тезисов и о требованиях к представляемым для публикации тезисам представлена на сайте ИНФО.

Подробную информацию о конкурсе вы можете найти на сайте ИНФО:

<http://infojournal.ru/competition/info-2017/>

Контакты Оргкомитета:

Телефон: +7 (495) 140-1986

E-mail: readinfo@infojournal.ru

<http://www.infojournal.ru/>



Дорогие друзья!

Журнал «Информатика в школе», который вы держите в руках, особенный — он посвящен 145-летию Московского педагогического государственного университета — одного из крупнейших и старейших педагогических вузов страны.

История университета началась в ноябре 1872 года, когда российский историк и общественный деятель Владимир Иванович Герье с одобрения императора Александра II основал Московские высшие женские курсы — первое учебное заведение в России, открывшее женщинам любого сословия доступ к высшему образованию на родине.

Сегодня в Московском педагогическом государственном университете обучается около 30 тысяч студентов из 75 регионов России и 45 зарубежных государств.

Важнейшее направление работы университета — подготовка педагогических кадров: МПГУ лидирует по качеству приема в бакалавриат на педагогические направления подготовки.

Подготовка будущих учителей информатики осуществляется в Московском педагогическом государственном университете в бакалавриате по профилям:

- «Информатика и экономика»;
- «Информатика и математика»;
- «Математика и информатика»;
- «Начальное образование и информатика»;
- «Физика и информатика»;
- «Технология и информатика».

Возможности дальнейшего профессионального развития учителя информатики обеспечиваются магистерскими программами:

- «Профильное и углубленное обучение информатике»;
- «Организация современной информационной образовательной среды»;
- «Робототехника и электроника в образовании».

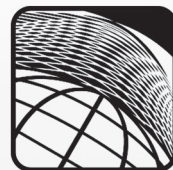
Ведется подготовка аспирантов и докторантов по направлению 44.06.01 «Образование и педагогические науки», направленность «Теория и методика обучения и воспитания (информатика)».

В 2005 году под руководством заслуженного деятеля науки Российской Федерации, лауреата премии Правительства Российской Федерации, академика РАО, доктора педагогических наук, профессора Александра Андреевича Кузнецова в МПГУ была создана кафедра теории и методики обучения информатике. В настоящее время на кафедре работают высококвалифицированные специалисты — доктора и кандидаты наук, научно-методические труды которых внедрены в практику обучения и оказывают существенное влияние на развитие методической системы обучения информатике в школе и вузе, на развитие современного цифрового образования. Руководит кафедрой Людмила Леонидовна Босова — заслуженный учитель Российской Федерации, лауреат премии Правительства Российской Федерации, доктор педагогических наук, автор учебников информатики для основной и старшей школы, главный редактор журнала «Информатика в школе».

На страницах этого номера журнала мы представляем статьи по методике обучения информатике не только известных ученых — преподавателей МПГУ, но и начинающих исследователей — аспирантов, магистрантов и студентов бакалавриата, предлагающих новые идеи, свежий взгляд на содержание подготовки школьников в области информатики и ИКТ.

Надеемся, что материалы данного выпуска журнала будут интересны и полезны вам, уважаемые читатели.

*Редакция журнала
«Информатика в школе»*



А. Г. Кушниренко,

Федеральный научный центр «Научно-исследовательский институт системных исследований»
Российской академии наук, г. Москва,

А. Г. Леонов,

Московский педагогический государственный университет

«КУМИР» — УЧЕБНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ БАЗОВОГО КУРСА ИНФОРМАТИКИ

Аннотация

В статье излагается история создания системы «КуМир», предназначенной для поддержки курса программирования в начальной и средней школе, а также рассмотрено место системы в современном курсе информатики.

Ключевые слова: алгоритм, программа, информация, информатика, «КуМир», базовый курс информатики.

Контактная информация

Кушниренко Анатолий Георгиевич, канд. физ.-мат. наук, доцент, зав. отделом учебной информатики Федерального научного центра «Научно-исследовательский институт системных исследований» Российской академии наук, г. Москва; адрес: 117218, г. Москва, Нахимовский пр-т, д. 36, корп. 1; телефон: (495) 718-21-10; e-mail: agk@mail.ru

Леонов Александр Георгиевич, канд. физ.-мат. наук, доцент, профессор кафедры математики и информатики в начальной школе Московского педагогического государственного университета; адрес: 119991, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; телефон: +7 (499) 245-03-10; e-mail: dr.l@infomir.ru

A. G. Kushnirenko,
Scientific Research Institute of System
Analysis of the Russian Academy of Sciences,
Moscow,

A. G. Leonov,
Moscow State University of Education

KUMIR — THE TEACHING SOFTWARE FOR BASIC INFORMATICS COURSE

Abstract

The article describes the history of the creation of the KuMir system to support the programming course in primary and secondary school, and also the place of the system in the modern course of informatics is considered.

Keywords: algorithm, program, information, informatics, KuMir, basic course of informatics.

1. Информатика — новый школьный предмет

Базовый курс информатики впервые появился в нашей стране (еще в СССР) в 1985 году, когда Постановлением ЦК КПСС и Совета Министров СССР № 271 от 28.03.85 «О мерах по обеспечению компьютерной грамотности учащихся средних учебных заведений и широкого внедрения электронно-вычислительной техники в учебный процесс» с 1 сентября того же года в девятых классах всех средних общеобразовательных школ было введено преподавание нового предмета «Основы информатики и вычислительной техники». Введенный в сжатые сроки курс информатики и ВТ использовал для изложения алгоритмов школьный алгоритмический язык, «отцом» которого является академик Андрей Петрович Ершов, автор и ответственный редактор первого отечественного учебника по информатике для школы [5].

Изучение основ алгоритмизации во вновь введенном курсе предполагало преподавание без компьютеров, как тогда говорили, безмашинное, поскольку за пять месяцев, отведенных для создания и внедрения курса, обеспечить все школы СССР компьютерами для проведения уроков информатики не было никакой возможности. Для машинного курса А. П. Ершов предполагал использовать разработанные ранее в Новосибирском государственном университете и Сибирском отделении АН СССР языки Робик и Рапира в составе учебной системы программирования «Школьница» [3, 6]. Разумеется, ряд решений, принятых в этих языках, А. П. Ершов использовал в школьном алгоритмическом языке. К сожалению, «Школьница» была реализована только на ЭВМ «Агат» и не получила большого распространения.

```
ФУНК СУММА (X) (*СУММА ВСЕХ ЭЛЕМЕНТОВ КОРТЕЖА*)
ИМЕНА: К
НАЧ ЗНАЧ := 0
ДЛЯ К ОТ 1 ДО ДЛИН(X) НЦ
  ЗНАЧ := ЗНАЧ + X [К]
КЦ
КОН
```

Рис. 1. Программа на языке Рапира — суммирование последовательности чисел

2. «Е-практикум» — компьютерная поддержка безмашинного курса

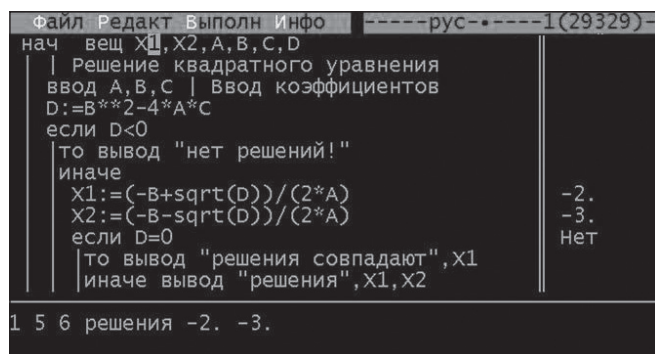
Исполнителям постановления ЦК и Совмина было ясно, что освоение школьного алгоритмического языка, на котором было построено пробное учебное пособие, потребует от педагогов значительных усилий.

Поэтому ведущим высшим учебным заведениям СССР, в том числе МГУ имени М. В. Ломоносова, была поставлена задача — в течение короткого периода летних каникул 1985 года подготовить методистов и преподавателей для ведения нового предмета в школах СССР. На курсы переподготовки были призваны педагоги (в основном математики), которые в большинстве своем ранее с программированием знакомы не были и не имели даже навыков написания простейших программ. За две недели их нужно было научить всему, что было написано в первой части учебника [5] (для девятого класса). Труднее всего было научить педагогов уверенно составлять простейшие алгоритмы на школьном алгоритмическом языке.

Задача была трудна не только методически, но и технически — за две недели требовалось проверить по несколько десятков программ, составленных каждым из нескольких сотен слушателей.

Разные учебные заведения решали эту задачу по-разному. На механико-математическом факультете МГУ коллективом студентов, аспирантов и сотрудников факультета под руководством А. Г. Кушниренко в рекордно короткие сроки был разработан и реализован редактор-компилятор со школьного алгоритмического языка. Это удалось сделать, используя накопленный опыт разработки редакторов-компиляторов для учебного процесса мехмата. Сегодня подобные системы называют IDE (Integrated Development Environment) — интегрированными средами разработки.

На мехмате к этому времени уже эксплуатировались учебные программные системы «Альфа-практикум» и «Бета-практикум», и потому новая система для школьной информатики в честь Ершова получила название «Е-практикум» [1]. Созданная система была впоследствии реализована на всех отечественных и зарубежных ЭВМ, которые использовались в советской, далее российской школе: «Корвет», УКНЦ, БК-0010, MSX-2 (Yamaha), Atari, IBM-PC и т. д.



```

Файл Редакт Выполни Инфо  ----рус----1(29329)
нач  вещь X1,X2,A,B,C,D
| Решение квадратного уравнения
ввод A,B,C | Ввод коэффициентов
D:=B*B-4*A*C
если D<0
| то вывод "нет решений!"
иначе
  X1:=(-B+sqrt(D))/(2*A)
  X2:=(-B-sqrt(D))/(2*A)
  если D=0
  | то вывод "решения совпадают",X1
  | иначе вывод "решения",X1,X2
1 5 6 решения -2. -3.
  
```

Рис. 2. Нахождение корней квадратного уравнения в «Е-практикуме»

3. «КуМир» — на пути к машинному курсу

В 1988 году коллектив авторов под руководством академика А. П. Ершова (включая сотрудников мехмата МГУ А. Г. Кушниренко и Г. В. Лебедева — авторов интерфейса системы «Е-практикум») выпускает новый учебник «Основы информатики и вычислительной техники» [4]. При написании учебника авторы пытались выдержать принцип: все алгоритмы, которые разбираются в учебнике, должны быть выполнимы в «Е-практикуме»,

и все задачи из учебника должны допускать решение в «Е-практикуме». В полной мере этот принцип выдержать не удалось, так как не хватало возможностей поддерживаемого «Е-практикумом» языка и отсутствовал необходимый набор исполнителей, которыми можно было бы управлять средствами языка. Стало ясно, что «Е-практикум» должен уступить место полноценной учебной системе со встроенными виртуальными исполнителями, включая Робота и Чертежника.

Такая система была разработана и реализована сотрудниками мехмата МГУ А. Г. Леоновым и М. Г. Эпикетовым и получила название «КуМир» — «Комплект учебных Миров». На первых порах в комплект входили Робот, Чертежник и знаменитая Черепашка С. Пейперта. «КуМир» разрабатывался параллельно с выпущенным в 1990 году издательством «Просвещение» учебником информатики [12] и был ориентирован на поддержку этого учебника. С 1990 по 1996 год учебник выдержал четыре издания с общим тиражом более семи миллионов экземпляров, а поддерживающая учебник система «КуМир» в 1995 году была рекомендована Министерством образования РФ в качестве базового программного обеспечения для школьного курса информатики.

В последнем десятилетии XX века учебник [12] и система «КуМир» продолжали использоваться во многих странах СНГ. Учебник был даже переведен на узбекский и молдавский языки, и «КуМир» был доработан для использования с этими учебниками. Тем временем информатика в школах РФ «спустилась» в VII—IX классы, и в 2000 году издательством «Дрофа» был выпущен учебник «Информатика. 7—9 классы» [11], также ориентированный на школьный алгоритмический язык и тем самым на систему «КуМир».

4. Новая реализация «КуМира» и новая редакция ФГОС основной школы

Выпущенный в 1990 году «КуМир» работал под управлением операционной системы MS DOS, которая в середине 90-х годов прошлого века уступила место оконным ОС семейств Windows и Linux. С каждым годом эксплуатировать «КуМир» на компьютерах с новыми ОС становилось все труднее, и интерфейс «КуМира» выглядел все более архаично.

Поэтому в начале XXI века под руководством А. Г. Леонова и М. А. Ройтберга система «КуМир» была перепроектирована и реализована как многоплатформенная. Реализация была выполнена в отделе учебной информатики Научно-исследовательского института системных исследований Российской академии наук Д. В. Хачко и В. В. Яковлевым. В новом «КуМире» появились новые исполнители, школьный алгоритмический язык был расширен с сохранением обратной совместимости.

Сегодня система может быть установлена на компьютерах с операционными системами семейств Windows, Linux и macOS [17].

Важным нововведением оказалась возможность разработки в «КуМире» практикумов по программированию с автоматизированной проверкой заданий. С помощью этого механизма был подготовлен ряд практикумов известными специалистами в методике преподавания школьной информатики [8, 10, 20].

Хотя скорость исполнения программ не очень важна при работе по обязательной программе по информатике,

для олимпиад необходим некоторый минимум скорости, и по скорости исполнения новый «КуМир» 2.x доведен до уровня языка Python.

Находящаяся в стадии утверждения новая редакция ФГОС основной школы по информатике [21] вводит ограничительный перечень языков программирования, допущенных к использованию в основной школе, и открывают этот перечень школьный алгоритмический язык и Pascal.

В 2016 году издательство «Дрофа» выпустило комплект учебников по информатике [13–15], в котором можно найти 171 законченный алгоритм на школьном алгоритмическом языке и пару десятков копий экранов системы «КуМир», показывающих результаты исполнения алгоритмов.

Сегодня готовится переработка этих учебников в соответствии с предстоящими изменениями в ФГОС основной школы.

5. Причины устойчивого интереса российских школ к «КуМиру»

Поиск в Интернете по ключевым словам «КуМир» и «информатика» показывает, что с годами популярность школьного алгоритмического языка и системы «КуМир» не уменьшилась. «КуМир» используется и для кружковой работы в V–VI классах, и для прохождения отдельных тем в обязательном курсе в VII–IX классах. Можно ожидать, что с выходом новой редакции ФГОС и модернизированного учебника «Дрофы» популярность возрастет.

По каким же причинам система «КуМир» получила такое устойчивое распространение, а школьный алгоритмический язык до сих пор является эффективным инструментом для изучения основ алгоритмизации в школе?

По этому поводу могут быть разные мнения. Если в начале 90-х годов XX века можно было предполагать, что популярность обеспечило просто удачное стечение обстоятельств, то сегодня стоит провести более содержательный анализ, чтобы понять, что эту популярность обеспечило.

До начала анализа необходимо явно назвать те области применения, в которых «КуМир» популярен.

«КуМир» — это учебная программная система для первого знакомства школьников с идеями программирования в обязательном курсе информатики в VII–IX классах или в дополнительном образовании для V–VI классов.

Готовящаяся к утверждению новая редакция ФГОС основной школы отводит на программирование около 25–30 часов и перечисляет ряд конкретных задач по программированию, которые обучаемые должны уметь решать по окончании курса. Возможное введение федеральных проверочных работ по ряду предметов в конце каждого учебного года может поставить перед школами (и учителями информатики) задачу за 25–30 уроков научить возможно большее количество учеников решать задачи по программированию, перечисленные в новой редакции ФГОС. Для этого потребуются система программирования, доступная к освоению семиклассниками и требующая минимальных затрат на такое освоение.

По мнению авторов настоящей статьи, именно легкостью освоения «КуМира» и объясняется его популярность. Эта легкость обеспечивается рядом полезных особенностей школьного алгоритмического языка и системы «КуМир».

6. Удачные особенности школьного алгоритмического языка, заложенные А. П. Ершовым

К этим особенностям в первую очередь относятся:

- 1) изначальная нацеленность на учебный характер языка и на достижение низкого порога вхождения;
- 2) русскоязычная лексика;
- 3) поддержка внешних исполнителей;
- 4) консервативность синтаксиса и семантики, близость к проверенному временем учебному языку Pascal.

Рассмотрим каждый из этих пунктов подробнее.

6.1. Одной из трудностей выбора языка для записи алгоритмов в школе А. П. Ершов называл противоречие между разнообразием языковой практики программирования и единством учебного процесса в школе. Использование производственных языков программирования в обязательном курсе информатики в VII–IX классах не представляется разумным, так как в нем не ставится задача подготовки профессиональных программистов.

Использование учебных языков, школьного алгоритмического языка и Pascal уменьшает технические затраты на освоение собственно языка и оставляет больше времени на алгоритмические вопросы. В легкости освоения школьный алгоритмический язык выигрывает у Pascal, особенно для русскоязычных учеников. На освоение основных конструкций школьного алгоритмического языка рядовой ученик тратит в разы меньше времени, чем пришлось бы потратить при изучении одного из производственных языков. Опыт одного из авторов, А. Г. Леонова, показывает, что изучить алгоритмический язык (в системе «КуМир») и освоить основы алгоритмизации можно всего за шесть часов [19]. Для курсов на основе производственного языка, скажем, C, подобный результат совершенно недостижим.

В учебнике по информатике и вычислительной технике [12] перечисление всех конструкций алгоритмического языка занимает всего половину форзаца. Редкий язык программирования может похвастаться такой лаконичностью.

6.2. Часто можно услышать мнение, что англоязычная лексика практически не мешает новичкам осваивать языки программирования. Это мнение не подтверждается практикой, и выбор русскоязычной лексики в школьном алгоритмическом языке заметно ускоряет освоение основ программирования.

6.3. После введения С. Пейпертом знаменитой Черепашки стало ясно, что освоение начал программирования легче проводить на примерах наглядных алгоритмов управления, а не на примерах более абстрактных алгоритмов вычисления или преобразования текстов. А. П. Ершов активно пропагандировал этот подход, и возможность управления внешними исполнителями была предусмотрена еще в языке Робик. Использование простых и наглядных исполнителей Робот и Чертежник во многом обеспечило успех учебников информатики [11, 12].

6.4. Несмотря на то что к 1985 году в языках Робик и Рапира был предложен ряд оригинальных синтаксических и архитектурных решений, в школьном алгоритмическом языке были использованы только консервативные решения. Язык получился похожим на Pascal (замечательное описание которого [7] было издано тремя годами

раньше), что способствовало популярности школьного алгоритмического языка среди школьных учителей.

7. Полезные расширения школьного алгоритмического языка, введенные при переходе от «Е-практикума» к «КуМиру»

Эти расширения можно рассматривать как материализацию философского замечания А. П. Ершова, высказанного им в статье 1985 года: «В отличие от жестких языков программирования алгоритмический язык обладает некоторой синтаксической свободой, присущей языку “деловой прозы”, ориентированной на читателя-человека» [2].

Об одном из проявлений такой свободы уже говорилось — это русскоязычная лексика: ключевые слова записываются кириллицей, в идентификаторах использование кириллицы разрешено. При этом русскоязычность не доводится до абсурда: традиционные функции $\sin$, $\cos$, $\log$ и др. используются в традиционном написании, имена, менее распространенные — $\bmod$ и $\div$, — также не русифицируются, в редакторе «КуМир»-программ латинские буквы по умолчанию выводятся курсивом, что снимает опасность смешения русских и латинских букв одного начертания.

Следующая важная особенность — возможность использовать в идентификаторах пробельные символы, т. е. использовать многословные идентификаторы. Согласитесь, что имя алгоритма «Идти вдоль стены» гораздо лучше смотрится (и читается), чем «Идти_вдоль_стены».

Возможность использования многословных идентификаторов немедленно влечет за собой необходимость введения еще одной возможности языка. Представьте себе, что требуется использовать отрицание значения логической величины с многословным именем, например, с именем «робот у стены». Грамматически правильное размещение ключевого слова «не» перед многословным именем логической переменной приводит к выражению «не робот у стены», что приводит к лингвистической катастрофе — с точки зрения русского языка это выражение имеет смысл, но совсем не тот, который нам нужен. Поэтому в школьном алгоритмическом языке разрешается размещать ключевое слово «не» как перед первым словом многословного имени, так и между любыми словами многословного имени. В рассматриваемом случае достаточно вместо «не робот у стены» написать «робот не у стены», и все встает на свои места: с точки зрения русского языка выражение «робот не у стены» является отрицанием выражения «робот у стены».

Естественность алгоритмического языка еще и в том, что семантика управляющих конструкций частично наследуется из естественного языка. В частности, конструкция ветвления «если — то — иначе — все» школьниками воспринимается легко, как пришедшая из повседневной жизни, в которой детям приходится выслушивать фразы типа: «ЕСЛИ ты не сделаешь уроки, ТО гулять не пойдешь». Как видно, запись конструкции ветвления не сильно отличается от того, что знакомо любому ученику с детства.

Доводы в пользу использования латиницы в записи конструкций легко парируются следующим шуточным вопросом: слышали ли вы хоть раз в жизни, чтобы отец или мать вдруг сказали своему ребенку: «If ты не сделаешь уроки then не пойдешь гулять».

Разумеется, в алгоритмическом языке присутствуют определенные алгебраичность и формализм. Например, в конструкции цикла используются короткие ключевые слова-сокращения «нц — кц», вместо длинных словосочетаний «начало цикла — конец цикла»:

```
алг К стене вправо
нач
. нц пока справа свободно
. . вправо
. кц
кон
```

Практика показывает, что дети мгновенно запоминают и легко осваивают подобные сокращения, понимая их необходимость.

Еще одна особенность школьного алгоритмического языка — это возможность использования принятой в математике формы записи сложных неравенств типа « $0 < x < 1$ » вместо несуразной записи « $0 < x$ и $x < 1$ ». В производственных языках программирования первым от этой несуразности избавился Python.

8. Полезные особенности редактора-компилятора системы «КуМир»

Еще одна особенность системы «КуМир», сближающая ее с Python, — сведение к нулю синтаксической значимости закрывающих операторных скобок. В принципе в «КуМир»-программе можно удалить все закрывающие операторные скобки и не потерять никакой информации о программе. Это достигается путем автоматической генерации вертикальной разметки и отступов в редакторе программ системы «КуМир».

Кроме того, можно легко исключить ошибки во вводе управляющих конструкций, выбирая из меню макрокоманды, вставляющие шаблоны конструкций целиком.

Фирменной особенностью «КуМира» является проверка корректности глобальной структуры программы в процессе редактирования программы. Полная диагностика всей программы, включая обнаружение неопределенных или повторно определенных имен, выполняется при любом покидании строки.

Очень важно, что «КуМир» подробно и в большинстве случаев удивительно точно диагностирует синтаксические ошибки, сигнализируя об ошибке как можно ближе к месту ее возникновения, выводя диагностику на поля программы и расцвечивая ошибочные строки. Это обеспечивается тем, что формальная грамматика языка в системе «КуМир» охватывает не только правильные программы, но часть программ, которые могут быть превращены в правильные исправлением одной ошибки.

9. Полезные особенности интерпретатора системы «КуМир»

Все одиночные величины и элементы массивов в момент их создания имеют специальное значение «не определено». Попытка использования неопределенной переменной или элемента массива диагностируется на этапе выполнения. Выходы индексов массива за допустимые границы также диагностируются.

В отличие от производственных языков программирования, выход за допустимый диапазон значений целочисленных переменных и элементов массивов в «КуМире» диагностируется на этапе выполнения. Поэтому попытка вычисления 31-й степени двойки в «КуМире»

приводит к диагностированному переполнению, в то время как, например, в Pascal ABC это вычисление дает отрицательное число.

В системе «КуМир» нет режима командной строки и выполнить вырванный из контекста фрагмент кода нельзя. Однако в «КуМире» можно без хлопот запустить на выполнение вспомогательный алгоритм с аргументами и результатами, не тратя время на составление основного алгоритма. При таком запуске аргументы будут запрошены до начала выполнения, а результаты будут выведены на экран по завершении выполнения. Эта особенность экономит много времени при тестировании несложных вспомогательных алгоритмов.

Еще одна фирменная особенность системы «КуМир» — отладочный режим, при котором результаты всех присваиваний и проверок условий выводятся на поля. Во многих случаях это избавляет новичка от использования отладочных печатей.

10. Отладочные возможности «КуМир»-интерпретатора

«КуМир» снабжен минимальным набором отладочных возможностей: есть точки останова, операторы «стоп» и «пауза», пошаговый режим исполнения, отображение в отладочном режиме результатов присваиваний на полях программы, режим отображения всех локальных и глобальных переменных и стека вызовов вспомогательных алгоритмов.

Из этого минимального набора новичок, как правило, использует при отладке только один удобный режим — пошаговое выполнение с выводом результатов присваиваний на поля.

11. «КуМир» не помогает придумать программу, но помогает записать и отладить то, что придумано

Начиная осваивать программистские премудрости, новичок должен для каждого нового задания одновременно преодолеть три трудности:

- придумать алгоритм;
- записать алгоритм на еще плохо освоенном языке программирования в еще плохо освоенной среде программирования;
- запустить свою программу, используя незнакомую среду программирования, понять, что эта программа делает правильно, а что нет, и попытаться исправить все ошибки.

В преодолении первой трудности «КуМир» мало чем может помочь.

Вторая трудность тем тяжелее, чем сложнее используемый язык программирования. Чем язык сложнее, тем проблематичнее без ошибок выразить на нем придуманный алгоритм. Профессиональный программист, накопив большой опыт, с проблемой «перевода» придуманного алгоритма на язык программирования, естественно, справляется лучше. Но для достижения такого уровня профессионал тратит годы, а в школьной программе на все программирование отводится 25–30 часов. Поэтому помочь преодолеть вторую трудность могут только такие особенности, как простота школьного алгоритмического языка, интеграция редактора и компилятора и общая ориентация «КуМира» на новичков.

Чтобы преодолеть третью трудность, новичок должен прежде всего «увидеть», что делает составленная им программа. На этом этапе помогают отладочные возможности интерпретатора «КуМира». Фирменный отладочный режим «КуМира» — пошаговое выполнение с выводом результатов на поля программы — замечательным тем, что не требует никакого изменения программы и может быть запущен в любой момент.

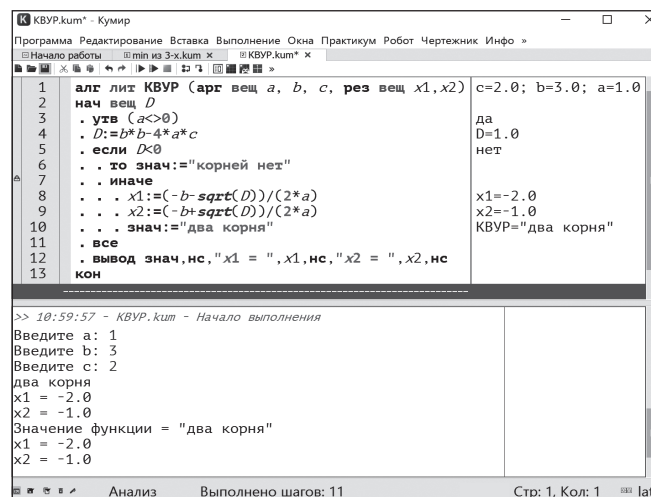


Рис. 3. Вспомогательный алгоритм нахождения корней квадратного уравнения тестируется без составления вызывающего алгоритма.

Значения аргументов запрашиваются, результаты выводятся на экран, и весь ход процесса выполнения алгоритма наглядно показан на полях программы

12. Возможности «КуМира» по автоматизированной проверке заданий на составление программ

Такие возможности сегодня есть практически во всех учебных системах. Что касается «КуМира», то его особенность в том, что проверяющая программа составляется на школьном алгоритмическом языке и автоматически интегрируется с проверяемой программой. Возможность предварительной проверки обучаемый получает офлайн, сразу после решения задачи. И только официальная сдача задания требует связи с Интернетом. Сегодня в открытом доступе есть несколько сотен заданий, обеспечивающих автоматическую проверку правильности их решения.

13. На каком языке программирования и в каком возрасте дети могут начинать знакомиться с программированием

Школьному алгоритмическому языку больше 30 лет. Программы учебника 1990 года [12] выполнимы в сегодняшней версии системы «КуМир». На этом основании иногда делается вывод, что школьный алгоритмический язык устарел.

Однако за эти 30 лет не претерпели существенных изменений и многие другие вещи: таблица умножения, формулы суммирования прогрессий, алгоритмы поиска наибольшего общего делителя, теорема Пифагора, классические версии языков C и Pascal.

И это говорит не об устаревании теоремы Пифагора или языков C и Pascal, а об их зрелости. Достигли опреде-

ленной зрелости и методика обучения основам школьной информатики с использованием школьного алгоритмического языка, предложенная академиком А. П. Ершовым, и программная поддержка этой методики. Для решения определенного круга задач данная методика и школьный алгоритмический язык остаются актуальными.

Другое дело, что оптимальный возраст первого знакомства с программированием с развитием информационных технологий и информационного общества продолжает понижаться. В наши дни знакомиться с программированием в седьмом классе, мягко говоря, поздно. Начинать знакомство с программированием лучше в начальной школе, а еще лучше до школы, в возрасте 6+. А это значит, что никакой текстовый язык и текстовая система программирования для первого знакомства не подходят. Это знакомство должно проводиться в пиктограммных системах программирования типа «ПиктоМир». Вместе с тем на роль первого текстового языка программирования для учащихся II—VI классов школьный алгоритмический язык по-прежнему может претендовать, принося в новую эпоху цифровой экономики и цифрового образования проверенную эффективную методику минувших лет [16].

За три десятилетия существования система «КуМир» претерпела многие изменения и коренные переработки. Часть из них коснулась эффективности и быстродействия самой системы, часть — интерфейса пользователя [18]. Все эти изменения были направлены на решение задачи, поставленной еще академиком А. П. Ершовым, — повышение удобства использования системы в реальном учебном процессе.

14. Обеспечение цифровой грамотности — задача национального уровня

В ходе выступления на Петербургском экономическом форуме в июне 2017 года Президент России Владимир Владимирович Путин отметил важность компетенций, которые дает школьный курс информатики, подчеркнув, что «по сути нам предстоит решить более широкую задачу, задачу национального уровня — добиться всеобщей цифровизации, цифровой грамотности».

Однако в современной сложной мировой геополитической обстановке ряд регионов России может столкнуться с трудностями при использовании импортного программного обеспечения. Не какого-то редкого и сверхсложного, а самого обыкновенного, повседневно используемого. Так, в Крыму и Севастополе не функционирует магазин приложений App Store компании Apple Inc. (США), являющийся практически единственным поставщиком приложений (в том числе учебных) для мобильных телефонов iPhone и планшетов iPad той же компании, а также для персональных компьютеров Mac.

Буквально недавно, 5 сентября 2017 года, сообщество Fedora Project, которое занимается разработкой открытой операционной системы Fedora Linux, изменило экспортное соглашение для пользователей, фактически запретив использование данной ОС в ряде районов мира. Опасность этого прецедента в том, что речь идет о свободном программном обеспечении [9].

Эти факты говорят в пользу максимального использования в России отечественного программного обеспечения.

Список использованных источников

1. Варсанофьев Д. В., Кушниренко А. Г., Лебедев Г. В. Е-практикум — программное обеспечение школьного курса информатики и вычислительной техники // Микропроцессорные средства и системы. 1985. № 3.
2. Ершов А. П. Алгоритмический язык в школьном курсе основ информатики и вычислительной техники // Микропроцессорные средства и системы. 1985. № 2.
3. Ершов А. П., Звенигородский Г. А. Урок 1: правила записи предписаний на языке Робик. Урок 2: гибкие системы предписаний, синтаксические диаграммы и переменные поля // Квант. 1979. № 9.
4. Ершов А. П., Кушниренко А. Г., Лебедев Г. В., Семёнов А. Л., Шень А. Х. Пробный учебник для средних учебных заведений. М.: Просвещение, 1988.
5. Ершов А. П., Монахов В. М., Бешенков С. А. и др. Основы информатики и вычислительной техники: пробное учебное пособие для средних учебных заведений. В 2 ч. / под ред. А. П. Ершова, В. М. Монахова. М.: Просвещение, 1985, 1986.
6. Звенигородский Г. А. Основные операторы учебно-производственного языка Рапира // Квант. 1980. № 1.
7. Йенсен К., Вирт Н. Паскаль. Руководство для пользователя и описание языка. М.: Финансы и статистика, 1982.
8. Кириенко Д. П. Курс алгоритмизации с использованием исполнителей системы КуМир и автоматического тестирования // Портал школы 179. <https://server.179.ru/wiki/?page=DenisKirienko/Kumir>
9. Коломыченко М. Свободную операционную систему Fedora Linux запретили поставлять в Крым // РБК. http://www.rbc.ru/technology_and_media/06/09/2017/59afcf49a7947349faa0ada
10. Кушниренко А. Г. Пять практикумов К. Ю. Полякова по программированию с автоматизированной проверкой в системе КуМир. Результаты переподготовки учителей информатики // Десятая конференция «Свободное программное обеспечение в высшей школе»: Тезисы докладов (г. Переславль, 24–25 января 2015 года). М.: Альт Линукс, 2015.
11. Кушниренко А. Г., Лебедев Г. В., Зайдельман Я. Н. Информатика. 7–9 классы. М.: Дрофа, 2000, 2002, 2003.
12. Кушниренко А. Г., Лебедев Г. В., Сворень Р. А. Основы информатики и вычислительной техники: пробный учебник для средних учебных заведений: учебное издание. М.: Просвещение, 1990, 1991, 1993, 1996.
13. Кушниренко А. Г., Леонов А. Г., Зайдельман Я. Н., Тарасова В. В. Информатика. 7 класс. М.: Дрофа, 2016.
14. Кушниренко А. Г., Леонов А. Г., Зайдельман Я. Н., Тарасова В. В. Информатика. 8 класс. М.: Дрофа, 2016.
15. Кушниренко А. Г., Леонов А. Г., Зайдельман Я. Н., Тарасова В. В. Информатика. 9 класс. М.: Дрофа, 2016.
16. Кушниренко А. Г., Леонов А. Г., Ройтберг М. А. Знакомим дошкольников и младших школьников с азами алгоритмики с помощью систем ПиктоМир и Кумир // Труды НИИСИ РАН. 2015. Т. 5. № 1.
17. Кушниренко А. Г., Леонов А. Г., Ройтберг М. А., Хачко Д. В., Яковлев В. В. Возможности системы КуМир // VI конференция «Свободное программное обеспечение в высшей школе» (г. Переславль, 30–31 января 2011 года): тезисы докладов. М.: Институт логики, 2011.
18. Кушниренко А. Г., Ройтберг М. А., Хачко Д. В., Яковлев В. В. Система программирования КУМИР 2.X // Труды НИИСИ РАН. 2015. Т. 5. № 1.
19. Леонов А. Г. Освой КуМир за 6 часов: последние четыре часа занятий // Информатика. 2011. № 2.
20. Поляков К. Ю. Практикумы в КуМир // Сайт К. Ю. Полякова. <http://kpolyakov.spb.ru/download/kumkurs.pdf>
21. Проект Приказа Министерства образования и науки РФ «Об утверждении федерального государственного образовательного стандарта основного общего образования в новой редакции» (подготовлен Минобрнауки России 09.07.2017). <https://www.garant.ru/products/ipo/prime/doc/56619643/>

Л. Л. Босова, Е. О. Сивкова,
Московский педагогический государственный университет

ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Аннотация

В статье рассмотрены планируемые результаты изучения курса информатики в основной школе, связанные с элементами теории множеств. Приведены основные теоретические сведения, с которыми следует познакомить школьников, и рассмотрены примеры решения задач. Предложена подборка задач для самостоятельного решения.

Ключевые слова: теория множеств, математические основы информатики, информатика в основной школе.

Контактная информация

Босова Людмила Леонидовна, заслуженный учитель РФ, доктор пед. наук, доцент, зав. кафедрой математики и информатики в начальной школе Московского педагогического государственного университета; *адрес:* 119991, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *телефон:* (499) 438-79-77; *e-mail:* akulll@mail.ru

Сивкова Елена Олеговна, канд. физ.-мат. наук, доцент кафедры математики и информатики в начальной школе Московского педагогического государственного университета; *адрес:* 119991, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *телефон:* (499) 438-79-77; *e-mail:* sivkova_elena@inbox.ru

L. L. Bosova, E. O. Sivkova,
Moscow State University of Education

ELEMENTS OF SETS THEORY IN THE SCHOOL COURSE OF INFORMATICS

Abstract

The article deals with the intended learning outcomes of the informatics course in secondary school related to the elements of sets theory. The main theoretical information with which students should be introduced is presented, and examples of solving problems are considered. A selection of tasks for individual work is offered.

Keywords: sets theory, mathematical foundations of informatics, informatics in secondary school.

Одна из проблем современного школьного образования — «изолированное» изучение учащимися системы научных понятий, составляющих содержание учебного предмета. Обучающиеся не только не могут использовать полученные на уроке знания и умения для решения жизненных задач, но и не «приучены» применять их в процессе учебно-познавательной деятельности по другим предметам. Основные причины этого явления кроются в том, что в учебных материалах по разным предметам отсутствуют горизонтальные взаимосвязи; провозглашенный на словах принцип реализации межпредметных связей на практике практически не встречается.

Наличие междисциплинарных связей между математикой и информатикой является общепризнанным, это зафиксировано в разного рода нормативных и методических документах федерального уровня. Так, в федеральном ядре содержания общего образования указывается: «Особого внимания заслуживают междисциплинарные связи математики и информатики. Это ни в коей мере не конкурирующие дисциплины (например, на почве компьютерного доказательства теорем или использования математических пакетов). При этом информатика — это не часть математики, хотя ряд понятий может быть одновременно отнесен к компетенции обеих дисциплин. Более продуктивно рассматривать математику и информатику как дисциплины, в определенной мере дополняющие друг друга» [7, с. 39]. «Математические основы информатики» — так называется один из четырех содержательных разделов современного школьного курса информатики, представленных в Примерной основной образовательной программе основного общего образования [6]. Изучение этого раздела предполагает рассмотрение систем счисления, элементов математической логики, комбинаторики, теории графов и теории множеств с точки зрения их применения в области информатики и информационных технологий.

Что касается элементов теории множеств (понятие множества, операции объединения и пересечения), то они в неявном виде присутствуют в курсе информатики достаточно давно и даже включены в государственную итоговую аттестацию по информатике (задание 18 в ОГЭ и задание 17 в ЕГЭ), где используются в контексте поисковых запросов.

Сегодня в качестве одного из планируемых предметных результатов изучения курса информатики в основной школе, согласно действующей версии Примерной основной образовательной программы основного общего образования, является умение «определять количество элементов в множествах, полученных из двух или трех базовых множеств с помощью операций объединения, пересечения и дополнения», достигаемое в процессе изучения следующего содержания: «Множество. Определение количества элементов во множествах, полученных из двух или трех базовых множеств с помощью операций объединения, пересечения и дополнения» [6].

В связи с тем, что данный материал представлен не во всех школьных учебниках информатики, а в курсе математики изучается фрагментарно и в более раннее время [3], **приведем в данной статье основные теоретические сведения и рассмотрим примеры задач, которые можно предложить ученикам основной школы на уроках информатики.**

Основные теоретические сведения

Множество — это совокупность объектов, объединенных некоторыми общими признаками.

Множества, содержащие конечное число элементов, называют *конечными* множествами.

Множество, не содержащее элементов, называется *пустым* множеством и обозначается символом $\emptyset$.

Множества, содержащие бесконечное число элементов, называют *бесконечными* множествами.

Множества M и P называются *равными* (обозначение $M = P$), если они состоят из одних и тех же элементов либо оба являются пустыми.

Если каждый элемент множества P принадлежит множеству M , то говорят, что P есть *подмножество* M и пишут $P \subset M$.

Объединение (сумма) множеств A и B (обозначается $A \cup B$) — это множество, элементы которого принадлежат хотя бы одному из множеств A или B .

Пересечение (произведение) множеств A и B (обозначается $A \cap B$) — это множество, элементы которого принадлежат одновременно множеству A и множеству B .

Работая с объектами какой-то определенной природы, всегда можно выделить в некотором смысле «самое большое» множество. Его называют *универсальным* множеством или *универсумом*. Так, рассматривая различные множества, состоящие из целых чисел, в качестве универсума можем взять множество Z . Обсуждая действительные решения различных уравнений и неравенств, универсумом обычно считают множество R . Универсальное множество принято обозначать буквой U .

Дополнение множества A (обозначается $\bar{A}$) — это множество элементов универсума U , не принадлежащих A .

Разность множеств A и B (обозначается $A \setminus B$) — это множество элементов из A , не принадлежащих B .

Симметрическая разность множеств A и B (обозначается $A \Delta B$) — это множество, являющееся объединением множеств $A \setminus B$ и $B \setminus A$.

Из школьного курса математики всем хорошо известны свойства арифметических операций (коммутативность, ассоциативность и др.). Подобными свойствами обладают также и некоторые операции над множествами.

Пусть A, B, C — подмножества универсального множества U . Для операций объединения, пересечения и дополнения справедливы следующие соотношения — *законы теории множеств*:

- законы коммутативности:
 $A \cup B = B \cup A$,
 $A \cap B = B \cap A$;
- законы ассоциативности:
 $A \cup (B \cup C) = (A \cup B) \cup C$,
 $A \cap (B \cap C) = (A \cap B) \cap C$;
- законы дистрибутивности:
 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$,
 $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$;
- законы тождества:
 $A \cup \emptyset = A$,
 $A \cap U = A$;
- законы дополнения:
 $A \cap \bar{A} = \emptyset$,
 $A \cup \bar{A} = U$;
- законы идемпотентности:
 $A \cup A = A$,
 $A \cap A = A$;
- закон двойного дополнения:
 $\bar{\bar{A}} = A$;
- законы де Моргана:
 $\overline{A \cup B} = \bar{A} \cap \bar{B}$,
 $\overline{A \cap B} = \bar{A} \cup \bar{B}$;
- законы поглощения:
 $(A \cup B) \cap A = A$,
 $(A \cap B) \cup A = A$.

Количество элементов в конечном множестве A называется *мощностью* этого множества и обозначается $|A|$.

Формула включений-исключений позволяет вычислить мощность объединения (пересечения) множеств, если известны мощности этих множеств и мощности всех их пересечений (объединений). Так, согласно этой формуле, для двух множеств имеет место соотношение:

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

Задачи

Рассмотрим несколько примеров.

Задача 1.

В классе 35 учеников, 20 из них занимаются в математическом кружке, 11 — в биологическом, а 10 ничем не занимаются.

Сколько ребят занимаются и математикой, и биологией?

Сколько ребят занимаются только математикой?

Сколько ребят занимаются только биологией?

Решение.

Обозначим через U множество всех учеников класса, а через M и B — множества учеников, занимающихся в математическом и биологическом кружках соответственно. Тогда

$M \cup B$ — множество учеников, которые занимаются, по крайней мере, в одном из кружков;

$M \cap B$ — множество учеников, которые занимаются в **обоих** кружках;

$\overline{M \cup B}$ — множество учеников, которые не занимаются ни в одном из кружков;

$M \cap \bar{B}$ — множество учеников, которые занимаются только математикой;

$B \cap \bar{M}$ — множество учеников, которые занимаются только биологией.

По условию:

$$|U| = 35;$$

$$|M| = 20;$$

$$|B| = 11;$$

$$|\overline{M \cup B}| = 10.$$

Найдем количество учеников, которые занимаются хотя бы в одном из кружков:

$$|U| = |M \cup B| + |\overline{M \cup B}|;$$

$$35 = |M \cup B| + 10;$$

$$|M \cup B| = 25.$$

Теперь найдем, сколько учеников занимаются в **обоих** кружках:

$$|M \cup B| = |M| + |B| - |M \cap B|;$$

$$25 = 20 + 11 - |M \cap B|;$$

$$|M \cap B| = 6.$$

Найдем количество учеников, которые занимаются только математикой:

$$|M| = |M \cap B| + |M \cap \bar{B}|;$$

$$20 = |M \cap \bar{B}| + 6;$$

$$|M \cap \bar{B}| = 14.$$

И, наконец, найдем, сколько учеников занимаются только биологией:

$$|B| = |B \cap \bar{M}| + |M \cap B|;$$

$$11 = |B \cap \bar{M}| + 6;$$

$$|B \cap \bar{M}| = 5.$$

На рисунке приведена диаграмма Эйлера—Венна, иллюстрирующая наше решение:

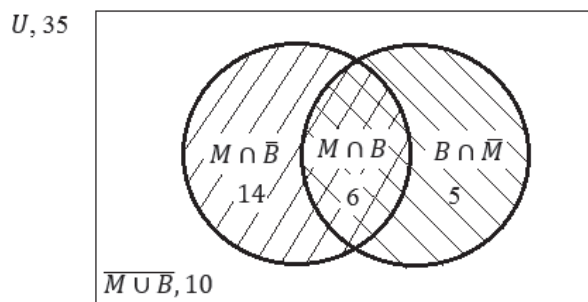


Рис. Графическая иллюстрация задачи 1

Задача 2 (Московская математическая олимпиада, 1938 год).

Сколько существует натуральных чисел, меньших тысячи, которые не делятся ни на 5, ни на 7?

Решение.

Обозначим через A и B множества натуральных чисел, меньших 1000, делящихся на 5 и на 7 соответственно. Тогда

- $A \cap B$ — множество натуральных чисел, меньших 1000, которые делятся и на 5, и на 7;
- $A \cup B$ — множество натуральных чисел, меньших 1000, которые делятся на 5 или на 7;
- $\overline{A \cap B}$ — множество натуральных чисел, меньших 1000, которые не делятся ни на 5, ни на 7.

Согласно закону де Моргана:

$$\overline{A \cap B} = \overline{A \cup B}.$$

Найдем мощности всех перечисленных множеств. Поскольку есть всего 999 натуральных чисел, меньших 1000 и среди них делится на 5 каждое пятое, то:

$$|A| = \left\lceil \frac{999}{5} \right\rceil = 199.$$

Аналогично:

$$|B| = \left\lceil \frac{999}{7} \right\rceil = 142.$$

Поскольку 5 и 7 — взаимно простые числа, то элементы множества $A \cap B$ должны делиться на 35. Поэтому:

$$|A \cap B| = \left\lceil \frac{999}{35} \right\rceil = 28.$$

Теперь по формуле включений-исключений можем найти:

$$|A \cup B| = 199 + 142 - 28 = 313.$$

Тогда:

$$|\overline{A \cap B}| = 999 - 313 = 686.$$

Итак, существует 686 натуральных чисел, меньших тысячи, которые не делятся ни на 5, ни на 7.

Задача 3 (олимпиада «Высшая проба», 2014 год).

В детский сад завезли карточки для обучения чтению. На некоторых написано МА, на остальных — НЯ. Каждый ребенок взял три карточки и стал составлять из них слова. Оказалось, что слово МАМА могут сложить из своих карточек 25 детей, слово НЯНЯ — 30 детей, слово МАНЯ — 36 детей. У скольких ребят все три карточки одинаковы?

Решение.

Возможны только четыре вида наборов из трех карточек. Пусть A , B , C и D — множества детей, имеющих карточки $\{MA, MA, MA\}$, $\{MA, MA, NY\}$, $\{MA, NY, NY\}$ и $\{NY, NY, NY\}$ соответственно. Тогда:

- $A \cup B$ — множество детей, которые могут сложить из своих карточек слово МАМА, и $|A \cup B| = 25$;
- $B \cup C$ — множество детей, которые могут сложить из своих карточек слово МАНЯ, и $|B \cup C| = 36$;
- $C \cup D$ — множество детей, которые могут сложить из своих карточек слово НЯНЯ, и $|C \cup D| = 30$;
- $A \cup D$ — множество детей, у которых карточки одинаковы.

Очевидно, что множества A , B , C и D попарно не пересекаются. Отсюда:

$$|A| + |B| = |A \cup B| = 25;$$

$$|B| + |C| = |B \cup C| = 36;$$

$$|C| + |D| = |C \cup D| = 30.$$

Требуется найти:

$$|A \cup D| = |A| + |D|.$$

Из трех записанных выше равенств следует:

$$|A| + |D| = (|A| + |B|) - (|B| + |C|) + (|C| + |D|) = 25 - 36 + 30 = 19.$$

Итак, три карточки одинаковы у 19 детей.

Задача 4 (олимпиада «Физтех», 2014 год).

На данный момент в классе 20 учеников, получивших сначала учебного года хотя бы одну двойку, 17 учеников, получивших не менее двух двоек, 8 учеников, получивших не менее трех двоек, 3 ученика, получивших не менее четырех двоек, и 1 ученик, получивший пять двоек. Больше пяти двоек нет ни у кого. Сколько всего двоек в журнале?

Решение.

Обозначим через A , B , C , D и E множества учеников, имеющих одну, две, три, четыре и пять двоек соответственно. Тогда

- $D \cup E$ — множество учеников, получивших не менее четырех двоек;
- $C \cup D \cup E$ — множество учеников, получивших не менее трех двоек;
- $B \cup C \cup D \cup E$ — множество учеников, получивших не менее двух двоек;
- $A \cup B \cup C \cup D \cup E$ — множество учеников, получивших хотя бы одну двойку.

Поскольку множества A , B , C , D и E попарно не пересекаются, то можем записать:

$$|D \cup E| = |D| + |E|,$$

$$3 = |D| + 1, |D| = 2;$$

$$|C \cup D \cup E| = |C| + |D \cup E|,$$

$$8 = |C| + 3, |C| = 5;$$

$$|B \cup C \cup D \cup E| = |B| + |C \cup D \cup E|,$$

$$17 = |B| + 8, |B| = 9;$$

$$|A \cup B \cup C \cup D \cup E| = |A| + |B \cup C \cup D \cup E|,$$

$$20 = |A| + 17, |A| = 3.$$

Отсюда общее количество двоек в журнале:

$$|A| + 2 \cdot |B| + 3 \cdot |C| + 4 \cdot |D| + 5 \cdot |E| = 3 + 18 + 15 + 8 + 5 = 49.$$

Задачи для самостоятельного решения

В заключение приведем подборку задач, которые могут быть предложены ученикам для самостоятельного решения. При составлении данной подборки использовались школьные учебники математики [3], учебники [2, 5] и учебные пособия [1] по информатике, материалы Московской математической олимпиады, олимпиад МГУ, Физтеха, ВШЭ [4].

1. Каждый ученик класса побывал в театре или в кино. В театр сходили 22 человека. В кино были 15 человек. И в театре, и в кино были 7 человек. Сколько учеников в классе? (30.)

2. В поход ходили 80 % учеников класса, а на экскурсии было 60 % класса, причем каждый был в походе или на экскурсии. Сколько процентов учеников класса были и в походе, и на экскурсии? (40.)

3. При опросе 100 семей выяснилось, что у 78 из них есть компьютер, у 85 — телевизор, а у 8 семей нет ни компьютера, ни телевизора. У скольких семей есть и компьютер, и телевизор? (71.)

4. По результатам опроса 52 школьников было установлено, что 23 из них собирают значки, 35 собирают марки, а 16 — и значки, и марки. Остальные не увлекаются коллекционированием. Сколько человек не увлекаются коллекционированием? (8.)

5. Из 80 туристов, приехавших в Москву, 52 хотят посетить Большой театр, 30 — Малый театр, 12 хотят посетить оба театра, остальные в театры ходить не хотят. Сколько человек не собираются идти в театр? (10.)

6. (Московская математическая олимпиада, 2006 год.) В саду у Ани и Вити росло 2006 розовых кустов. Витя полил половину всех кустов, и Аня полила половину всех кустов. При этом оказалось, что ровно три куста, самые красивые, были политы и Аней, и Витей. Сколько розовых кустов остались неполитыми? (3.)

7. Сколько существует натуральных чисел от 1 до 1000, которые не делятся ни на 2, ни на 5? (400.)

8. (Олимпиада «Ломоносов», 2012 год.) Сколько чисел из набора 1, 2, ..., 2010, 2011 не делятся ни на 3, ни на 7? (1149.)

9. Известно количество страниц, которые находит поисковый сервер по следующим запросам:

- Новосибирск и (Красноярск и Хабаровск или Норильск) — 570;
- Новосибирск и Норильск — 214;
- Новосибирск и Красноярск и Хабаровск и Норильск — 68.

Какое количество страниц будет найдено по запросу: «Новосибирск и Красноярск и Хабаровск»? (424.)

10. В целях стимулирования продаж магазин установил 10-процентную скидку на каждую пятую продаваемую газонокосилку и 25-процентную скидку на каждую тринадцатую продаваемую газонокосилку. В случае, если на одну газонокосилку выпадают обе скидки, применяется бóльшая из них. Всего в ходе рекламной акции было продано 1000 газонокосилок. Определите выручку магазина от продажи партии газонокосилок, если цена одной газонокосилки составляет 10 000 рублей. (9 625 000.)

11. Пусть A — множество букв слова «абракадабра», B — множество букв слова «бригантина», C — множество

букв слова «каракатица». Выпишите множества A , B и C , найдите их всевозможные пересечения и проверьте формулу включений и исключений для трех множеств.

12. Пол комнаты площадью 6 м² покрыт тремя коврами, площадь каждого из которых равна 3 м². Докажите, что какие-то два из этих ковров перекрываются по площади, не меньшей 1 м².

13. Сколько существует натуральных чисел от 1 до 900, которые не делятся ни на 2, ни на 3, ни на 5? (240.)

14. В первую смену в лагере «Дубки» отдыхали 30 отличников, 28 победителей олимпиад и 42 спортсмена. При этом 10 человек были и отличниками, и победителями олимпиад, 5 — отличниками и спортсменами, 8 — спортсменами и победителями олимпиад, 3 — и отличниками, и спортсменами, и победителями олимпиад. Сколько ребят отдыхало в лагере? (80.)

15. На пикник поехали 92 человека. Бутерброды с колбасой взяли 50 человек, с сыром — 60 человек, с ветчиной — 40 человек, с сыром и колбасой — 30 человек, с колбасой и ветчиной — 15 человек, с сыром и ветчиной — 25 человек, 5 человек взяли с собой все три вида бутербродов, а несколько человек взяли пирожки. Сколько человек взяли с собой пирожки? (7.)

16. В коллективе 70 человек. 25 из них знает английский язык, 16 — немецкий, 18 — французский, 5 — английский и немецкий, 7 — английский и французский, 3 — немецкий и французский, 3 — все три языка. Сколько людей знают хотя бы один из трех языков? (47.)

17. В группе 40 туристов. Из них 20 человек говорят по-английски, 15 — по-французски, 11 — по-испански. Английский и французский знают семь человек, английский и испанский — пятеро, французский и испанский — трое. Два туриста говорят на всех трех языках.

- 1) Сколько человек из группы не знают ни одного из этих языков?
 - 2) Сколько человек из группы знают только один язык?
 - 3) Сколько человек из группы знают любые два языка из трех?
 - 4) Сколько человек из группы говорят по-английски, а также по-французски или по-испански?
- (1) 7; 2) 22; 3) 9; 4) 10.)

18. Старшеклассники заполняли анкету с вопросами об экзаменах по выбору. Оказалось, что выбрали они информатику, физику и обществознание. В классе 38 учеников. Обществознание выбрал 21 ученик, причем трое из них выбрали еще и информатику, а шестеро — физику. Один ученик выбрал все три предмета. Всего информатику выбрали 13 учеников, пятеро из которых указали в анкете два предмета. Надо определить, сколько учеников выбрали физику. (15.)

19. (Олимпиада «Физтех», 2013 год.) Трое ребят принялись красить лист ватмана, каждый в свой цвет. Один закрасил красным 75 % листа, второй закрасил зеленым 70 % листа, а третий закрасил синим 65 % листа. Сколько процентов листа будет заведомо покрашено всеми тремя цветами? (10.)

20. Антон, Артем и Вера решили вместе 100 задач по математике. Каждый из них решил 60 задач. Назовем задачу трудной, если ее решил только один человек, и легкой, если ее решили все трое. На сколько отличается количество трудных задач от количества легких? (На 20.)

Список использованных источников

1. Андреева Е. В., Босова Л. Л., Филина И. Н. Математические основы информатики. Элективный курс: учебное пособие: 2-е изд., испр. М.: БИНОМ. Лаборатория знаний, 2007.
2. Босова Л. Л., Босова А. Ю. Информатика. 10 класс: учебник. М.: БИНОМ. Лаборатория знаний, 2016.
3. Дорофеев Г. В., Шарыгин И. Ф., Суворова С. Б. и др. Математика. 6 класс: учебник для общеобразовательных организаций: 5-е изд., дораб. М.: Просвещение, 2017.
4. Подготовка к олимпиадам и ЕГЭ по математике и физике. <http://mathus.ru/>

5. Поляков К. Ю., Еремин Е. А. Информатика. 10 класс. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 1. М.: БИНОМ. Лаборатория знаний, 2013.

6. Примерная основная образовательная программа основного общего образования. Одобрена решением федерального учебно-методического объединения по общему образованию (протокол от 8 апреля 2015 г. № 1/15). [http://минобрнауки.рф/документы/938/файл/4587/РООП_ООO_reestr_2015_01.doc](http://минобрнауки.рф/документы/938/файл/4587/РООП_ООО_reestr_2015_01.doc)

7. Фундаментальное ядро содержания общего образования / Рос. акад. наук, Рос. акад. образования; под ред. В. В. Козлова, А. М. Кондакова. 4-е изд., дораб. М.: Просвещение, 2011.

КОНКУРСЫ

В оформлении обложки данного номера журнала «Информатика в школе» использованы работы следующих авторов — победителей конкурса цифровых изображений и фотографий:



Полина Юдина,
ученица средней
общеобразовательной
школы № 1699
Управления делами
Президента Российской
Федерации, г. Москва



Дарья Галкина,
воспитанница ГОБУ ДО
«Дворец творчества —
Мемориал», ученица средней
общеобразовательной школы
№ 32, г. Киров



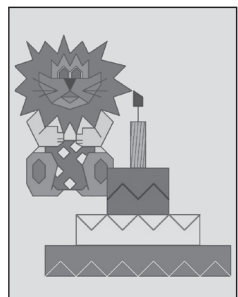
Иван Лебедев,
ученик Академического
лицея, г. Магнитогорск,
Челябинская область



Ольга Пудовкина,
ученица Академического
лицея, г. Магнитогорск,
Челябинская область



Екатерина Макшанцева,
ученица Академического
лицея, г. Магнитогорск,
Челябинская область



Елизавета Макагонова,
ученица Наро-Фомин-
ской средней общеобра-
зовательной школы № 1,
Московская область



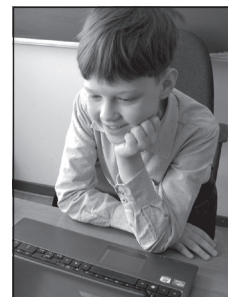
Яна Калинина,
студентка Барнауль-
ского государственного
педагогического
колледжа



С. А. Мячина,
учитель информатики
Борисоглебской гимназии
№ 1, Воронежская область



О. В. Степаненко,
учитель информатики
Борисоглебской гимназии
№ 1, Воронежская область



О. В. Ярошевич,
учитель информатики
гимназии
«Новое поколение»,
г. Москва

Уважаемые авторы и читатели!

Издательство «Образование и Информатика» объявляет о проведении конкурса цифровых изображений и фотографий, которые будут размещены на обложке журнала «Информатика в школе» в первом полугодии 2018 года.

Подробности — на сайте издательства «Образование и Информатика»: <http://www.infojournal.ru/>

Н. Н. Самылкина,
Московский педагогический государственный университет,

И. А. Калинин,
Государственный институт русского языка им. А. С. Пушкина, г. Москва

ВЛИЯНИЕ ОБРАЗОВАТЕЛЬНОЙ РОБОТОТЕХНИКИ НА СОДЕРЖАНИЕ КУРСА ИНФОРМАТИКИ ОСНОВНОЙ ШКОЛЫ

Аннотация

В статье обобщается влияние образовательной робототехники на содержание некоторых тем курса информатики основной школы. Рассматривается один из вариантов современного и компактного изложения таких тем, как «Архитектура компьютера» и «Программное обеспечение», в контексте изучения информационных систем.

Ключевые слова: робототехника, информатика, техносфера, информационная система, облачные сервисы.

Контактная информация

Самылкина Надежда Николаевна, канд. пед. наук, доцент, профессор кафедры теории и методики обучения информатике Московского педагогического государственного университета; *адрес:* 119991, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *телефон:* (499) 264-25-56; *e-mail:* nsamylkina@yandex.ru

Калинин Илья Александрович, канд. пед. наук, доцент, начальник управления информационных систем и сетей Государственного института русского языка им. А. С. Пушкина, г. Москва; *адрес:* 117485, г. Москва, ул. Академика Волгина, д. 6; *телефон:* (495) 330-88-38; *e-mail:* kalininilya@mail.ru

N. N. Samylkina,
Moscow State University of Education,
I. A. Kalinin,
Pushkin State Russian Language Institute,
Moscow

EDUCATIONAL ROBOTICS INFLUENCE ON THE CONTENT OF INFORMATICS COURSE IN SECONDARY SCHOOL

Abstract

The article generalizes the influence of educational robotics on the content of some topics of informatics course in secondary school. One of the versions of a modern and compact presentation of such topics as "Computer Architecture" and "Software" in the context of the study of information systems is considered.

Keywords: robotics, informatics, technosphere, information system, cloud services.

Образовательная робототехника в соответствии с Примерной основной образовательной программой основного общего образования является составной частью информатики [3]. За счет робототехники информатика в основной школе не просто получила в своем составе дополнительный раздел со сложным оборудованием и разными программными возможностями — робототехника существенно обновляет многие темы курса информатики. В последнее время даже при отсутствии робототехнического оборудования в школе многие ребята занимались робототехникой или программированием в различных центрах дополнительного образования. При наличии такого опыта у детей возникает множество вопросов, связанных с современными возможностями компьютеров, с актуальными направлениями ИТ-отрасли. Эти реалии необходимо будет учесть учителю информатики в самом ближайшем будущем. В предлагаемом материале мы остановимся именно на тех моментах, которые потребуют у учителя дополнительной подготовки при изучении некоторых тем курса информатики в основной школе помимо программирования.

Для современных учащихся именно школьная информатика открывает профессиональную дорогу в одну из динамично развивающихся технологических отраслей — отрасль информационных технологий. Информационные технологии как отрасль профессиональной деятельности представляют собой передний край тех изменений, которые происходят в обществе в целом, включая образование. *Отрасль занимается созданием, эксплуатацией и развитием информационных систем (ИС) в самых разных областях деятельности* [1]. Развиваясь за счет новых высокотехнологичных решений, ИТ-отрасль оказывает при этом серьезное влияние на все остальные сферы деятельности человека. Все инновационные решения в других отраслях возникают при взаимодействии друг с другом и на стыке с ИТ.

Развитие современных технологий во всех отраслях характеризуется:

- ростом взаимосвязей между технологическими отраслями за счет телекоммуникационных решений (увеличение объема данных и решений по их обработке);
- мобильностью и дружелюбностью цифровых решений (уменьшение размеров цифровых устройств и высокая концентрация возможностей в одном устройстве);
- новыми «умными средами» самого разного назначения (для работы, досуга, личного самоконтроля — браслеты, очки, кошельки и пр.).

Современные научные знания (открытия, изобретения, законы) и практический опыт, ставшие информационными ресурсами общества, можно эффективнее использовать за счет применения информационных технологий. Благодаря этому можно значительно сэкономить на других видах ресурсов — финансовых, энергетических, людских — и существенно ускорить по времени все процессы, в которых информационный обмен играет важную роль (бизнес-процессы, образовательные, процессы внедрения изобретений в практику, процессы внедрения Интернета вещей и пр.). Во всех областях деятельности человека стремительно создается *высокотехнологичная профессиональная среда, называемая **техносферой***, рассчитанная на высокоинтеллектуальный человеческий ресурс.

Робототехника и смежные с ней области — важная часть техносферы уже сейчас, а тем более в будущем. Информационные системы, ориентированные на использование в указанных областях, — центральная задача организации производства, логистики, научных исследований и т. д. Таким образом, *информационные системы должны включать в себя элементы робототехники.*

Учитывая, что все будут взаимодействовать в жизни и в профессии с информационными системами, начинать их изучать надо как можно раньше. Это не сложно сделать, если начинать с состава и функций каждого из компонентов ИС. Вместе с тем сформируется и понимание места робототехники и ее влияния на информатику.

Предлагаем один из вариантов современного и компактного изложения тем «Архитектура компьютера» и «Программное обеспечение», в котором изучение данных тем интегрировано в контекст изучения темы «Информационные системы».

Сначала уточним схему состава информационных систем (рис. 1). Проще всего выделить состав из определения ИС.

Понятие информационной системы может использоваться в двух смыслах:

- *информационная система в широком смысле этого слова* — совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для удовлетворения заданной информационной потребности группы людей — например, регистрация и учет товаров на складе, расчет маршрутов по карте, поиск и изучение документов и т. д.;
- *информационная система в узком смысле* — совокупность аппаратных и программных компонентов, решающих задачу удовлетворения информационной потребности [3].

В обоих случаях речь идет о том, что, для того чтобы удовлетворить потребность в некоторой информации (т. е. в ее накоплении, хранении, поиске и т. д.), необходимо создать целый комплекс средств, который такую задачу будет решать. Конкретная структура любой информационной системы будет определяться ее задачами.

Здесь важно понимать, что изучать устройство компьютера и программное обеспечение вне информационной системы в настоящее время бессмысленно. Отдельно от сети и ИС компьютер не является эффективным средством работы с информацией.

На схеме, изображенной на рисунке 1, видно заметное расширение состава аппаратных средств (помимо компьютера в сборке и системной шины как его центрального конструктивного решения), которые традиционно рассматривались в курсе информатики основной школы, — к аппаратной части относятся теперь микроконтроллер, всевозможные датчики и средства ввода-вывода.

Знакомство с аппаратной частью ИС начинается с объяснения функционала всех составляющих компонентов. Каждый из этих компонентов может также быть самостоятельным устройством (например, датчик со встроенными средствами обработки данных).

- **Компьютер** — универсальное вычислительное устройство, чаще всего с вариативной конфигурацией в зависимости от задачи. Компьютер состоит из блоков, которые конструктивно часто выполняются в виде отдельных плат (память, устройства ввода-вывода) или микросхем (системная шина). Изменить конфигурацию компьютера можно.
- **Датчик** — техническое средство для оценки изменений окружающей среды, которое вырабатывает сигналы, оптимальные для обработки, передачи и хранения в технической системе.



Рис. 1. Состав информационных систем

- **Микроконтроллер** — частный, но очень важный случай компьютера — это компьютер, содержащий несколько логических устройств внутри одной микросхемы. На микроконтроллерах строятся роботы, средства управления процессами, различные мелкие средства автоматизации. Изменить конфигурацию микроконтроллера нельзя.
- **Средства ввода и вывода** — устройства, позволяющие в той или иной форме вводить информацию (т. е. уже готовую, фактически — команды) и выводить ее (т. е. демонстрировать оператору). Эти устройства не обязательно постоянно входят в аппаратную часть, например, промышленной системы — но, как правило, присутствуют в ней. Это дисплеи, кнопочные панели, индикаторы и пр.

Расширим схему по аппаратной составляющей, раскрыв применение микроконтроллеров (рис. 2). Именно здесь появляются роботы — автономные и в составе автоматизированных комплексов.

В этой упрощенной схеме ничего не сказано о форме, в которой могут существовать представленные на схеме устройства. А форма может быть очень различной: части микросхемы, отдельная микросхема, блок или плата, внешнее устройство.

Микроконтроллеры являются основой автономных роботов и роботизированных комплексов.

Далее обобщаем уже известные обучающимся из робототехнических занятий возможности автономных роботов:

- получение и обработка информации об окружающей среде;



Рис. 2. Состав аппаратной части ИС

- длительная работа без вмешательства человека (автономность);
- возможность перемещения полностью или частично для манипуляций с объектами;
- безопасность для человека.

Исходя из этого, получаем возможность естественным способом **рассмотреть понятие сигнала, его кодирования и интерпретации микроконтроллером робота**. Если в школе есть необходимое оборудование, то на этом этапе лучше начинать практическую сборку роботов и теорию в минимальном объеме рассматривать параллельно с практикой сборки и программирования автономных роботов. Именно в контексте практической работы происходит знакомство учащихся с основами электроники, включая составление электронных схем, рассмотрение устройства датчиков и их функционирования (т. е. как происходит преобразование аналоговых сигналов в цифровые и наоборот). Отдельно заметим, что такие занятия могут проходить и в пятом—седьмом классах, где лишней детализации при объяснении не требуется. Дети все «познают» руками при сборке.

Раскрытию профессиональных перспектив робототехнической отрасли поможет **представление архитектуры автономных роботов** (рис. 3).

Степень детализации при объяснении этого материала выбирает учитель, но даже общая схема позволяет понять, почему робототехника — это часть информатики. Раскрывая архитектуру автономных роботов, по сути, мы **переходим к теме моделирования**.

Манипуляторы (исполнительные устройства, эффекторы) — технологически сложные модели частей/органов реальных объектов, деятельность которых можно изменять в результате управляющего воздействия. Основы моделирования в курсе информатики основной школы за счет робототехники получают новое теоретическое и практическое наполнение.

Изучение работы различных датчиков позволит на качественно новом уровне рассмотреть **тему обработки и кодирования различных сигналов программными средствами**. Именно такой акцент при изучении темы «Сигналы и кодирование» характерен для информатики, он несколько отличается от того, что изучает физика (природу сигнала, его общие характеристики).

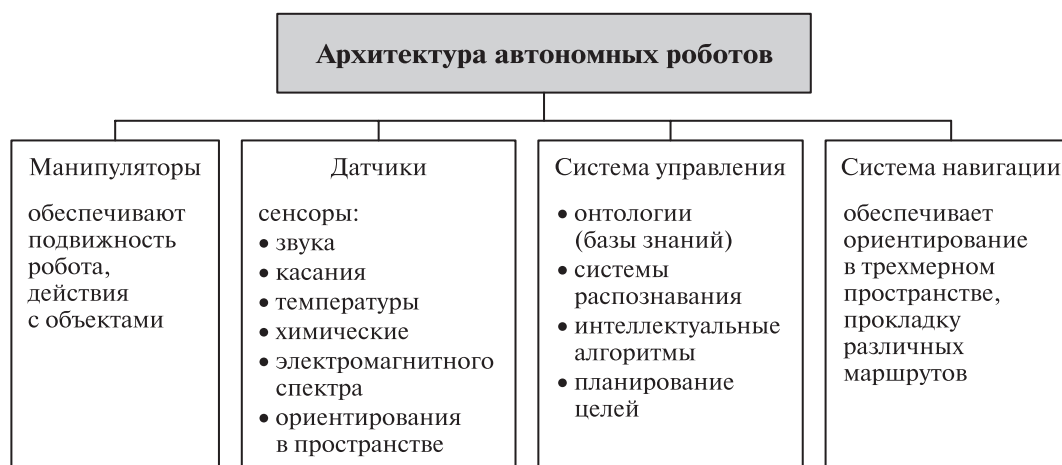


Рис. 3. Архитектура автономных роботов

Два блока из четырех на рисунке 3 (манипуляторы и датчики) тематически связаны с математикой, физикой и биологией.

Следующая тема, получающая новое звучание, — **алгебра логики**. Как раздел математики алгебра логики была одним из основных отправных пунктов не только при создании компьютеров, но и при создании первых программ, относящихся к искусственному интеллекту. Математическая логика до сих пор остается одной из основ создания и описания такого рода интеллектуальных систем. Основы логики становятся обязательными для изучения в основной школе. Причем от алгебры логики следует выйти на схемотехнику, т. е. непосредственную реализацию логических операций в различных микросхемах. Эта часть логической тематики в последнее время (начиная с ГОС-2004 и ФГОС ООО) стала незаслуженно игнорироваться, поскольку не выходит на итоговую аттестацию. Более того, в связи с ограниченным количеством часов на информатику, эта часть информатики, связанная со схемотехникой, с элементами бионики, может быть курсом по выбору в основной школе по двум причинам: *легко вычленивается в отдельный блок и не выходит на проверку в ходе итоговой аттестации*. Именно здесь впервые корректно возникает понятие «прототипирование» как важный этап создания электронного устройства.

Другие два блока, представленные на рисунке 3, — **система управления и система навигации** — позволяют понять возможности современных роботов, научиться управлять ими, расширять эти возможности. Это интеллектуальная составляющая роботов, реализованная в виде программного обеспечения. Специализированные программы позволяют получать и интерпретировать информацию от датчиков, управлять манипуляторами. Сложность окружающего мира заложена в предметные базы знаний (онтологии) с возможностью анализа в соответствии с правилами вывода, в системы распознавания речи, изображений, а также в системы с возможностью самообучения (системы машинного обучения, экспертные системы, нейронные сети и пр.). Именно здесь под общим названием **«искусственный интеллект»** сфокусированы все «цифровые навыки».

В настоящее время к задачам искусственного интеллекта относят, в частности [2]:

- *задачу представления знаний*, т. е. описания «понимания» человеком некоторой предметной области для последующего использования этого описания в автоматизированной обработке;
- *задачу машинного обучения*, т. е. выделения и записи общих закономерностей из наблюдений, организацию этих наблюдений. Люди учатся практически всю жизнь и обладают массой навыков, позволяющих им выделять общие закономерности;
- *задачу разработки средств машинного зрения*, т. е. установления соответствия между некоторым изображением и записью об объекте в базе данных. Человек (да и вообще живые существа) выполняет такое распознавание практически мгновенно. Технические системы — только в строго заданных узких ситуациях;
- *задачу обработки текстов на естественных языках*, т. е. возможность извлекать сведения и воспринимать команды на естественном человеческом языке;

- *задачу автоматического формирования логических выводов*;
- *задачу совершенствования средств робототехники*, т. е. манипуляции объектами в реальном пространстве, перемещений в пространстве.

Изучение тем искусственного интеллекта возможно в старшей школе, если в основной школе заложить основы моделирования и программирования на достаточном уровне, чтобы обучающиеся самостоятельно могли определиться с направлением в программировании, в котором они желают специализироваться в будущем.

Краткий вывод: робототехническая тематика в основной школе вырастает в направление «искусственный интеллект» в старшей школе.

Вторая часть состава ИС в виде программного обеспечения в теории дополнится **более подробным рассмотрением всего программного обеспечения и его классификаций**.

Наиболее частой и простой классификацией программного обеспечения является классификация ПО на **системное** (т. е. обеспечивающее управление и взаимодействие в целом) и **прикладное** (т. е. решающее конкретные задачи). Иногда из этой классификации выделяют также **утилиты** как средства решения узких задач, напрямую затрагивающих управление и взаимодействие, но не применяемых постоянно и автоматически и необязательных (например, средства диагностики и исправления ошибок файловых систем).

Средства, используемые для разработки информационных систем (в частности, включающих и обеспечивающих функционирование роботов и их комплексов), входят в состав обоих этих классов.

Например, к системному программному обеспечению относятся тесно интегрированные с ОС средства отладки (перехватывающие и контролирующие системные вызовы, состояние оперативной памяти и пр.).

К прикладному программному обеспечению могут быть отнесены специфические средства для создания программ на языках высокого уровня, средства взаимодействия с отладчиком, дополнительные программы для разработки ресурсов и описания средств взаимодействия (например, веб-страниц).

Некоторые компоненты современных информационных систем настолько велики, что включают в себя компоненты из всех слоев программного обеспечения, — например, это распространенные СУБД (Oracle, MS SQL, PostgreSQL и др.), в том числе собственные отдельные средства разработки.

В ряде случаев отдельные виды прикладного программного обеспечения могут быть интегрированы в обязательный состав и стать фактически частью ОС (например, СУБД, входящая в комплект мобильной ОС Android, хотя с точки зрения классического подхода большая часть Android — надстройка над ОС).

Средства разработки часто поставляются в составе крупных интегрированных систем, которые включают в себя все перечисленные компоненты.

Следует понимать, что классификация на системное и прикладное программное обеспечение, а также на «производящее» программное обеспечение и средства разработки достаточно условна — особенно для средств разработки (рис. 4). Многие компоненты поставляются как комплексы для разных задач, они могут быть доступны и целиком, и по частям и в процессе развития могут приобретать средства, которые ранее к ним никак не относились.



Рис. 4. Состав программного обеспечения

Например, современные браузеры содержат целый ряд средств разработки, особенно отладки содержимого веб-страниц, но при этом не выходят за рамки своей среды. При этом они могут использовать внешние компоненты, если они есть, или внешние компоненты могут использовать их как часть своей системы отображения через системные библиотеки, и т. д.

Развитие средств создания и обеспечения работы информационных систем, появление доступных и сравнительно дешевых средств хранения, памяти и, главное, высокоскоростных коммуникационных сетей привело к появлению такого направления создания программного обеспечения (и информационных систем, в частности), которое получило название **облачных технологий**, и концепции **«программное обеспечение как услуга»**. В настоящее время в учебниках информатики старшей школы только упоминают о возможностях облачных технологий. Но в самой ИТ-отрасли это направление предоставления услуг развивается стремительно. В ближайшее время интерес к этой теме у школьников будет возникать, вероятно, уже в основной школе. Поэтому возможен вариант пропедевтики темы облачных технологий в основной школе в контексте рассмотрения видов программного обеспечения и его функционирования.

В привычном, уже считающемся традиционным, способе организации информационной системы требуется подготовить аппаратную базу, развернуть необходимое системное и прикладное обеспечение, подготовить код и настроить систему, обучить персонал и работать, накапливая данные.

Для минимального случая такого подхода — персональной рабочей станции — потребуется приобрести компьютер и необходимое программное обеспечение, после чего, используя прикладные программы (например, офисное программное обеспечение), можно решать свои задачи. Возможности решения тех или иных задач зависят от имеющегося компьютера, его ресурсов и доступного программного обеспечения.

Конечно, в настоящее время невозможно представить компьютер, тем более в составе ИС, работающий вне **сети**.

Развитие средств работы с сетями позволило решать некоторые задачи принципиально по-другому — предоставляя новые возможности облачных сервисов как услугу. При этом подходе пользователь не знает, как именно организовано хранение, не занимается уста-

новкой и настройкой программного обеспечения — он решает только свою задачу.

Например, одна из самых простых задач — задача хранения своих файлов. Если хранить их на своем компьютере, то надо самому заботиться об их сохранности, о работе программного обеспечения, резервных копиях и доставке файлов в другие места. В случае использования облачного сервиса пользователь работает только с внешним интерфейсом сервиса (услуги) и получает доступ к файлам из любого места, где вообще есть доступ к Интернету. Все, что составляет техническую сторону сервиса, для пользователя не имеет структуры и воспринимается как облако.

Для доступа к таким услугам чаще всего используется браузер. Эти средства за последние 20 лет своего развития прошли путь от узкоспециализированных программ просмотра до крупных и многофункциональных комплексов, фактически позволяющих создавать специфические программы.

Таким образом, пользователю не требуется мощный ПК — достаточно планшета (или даже современного телевизора) и быстрого канала доступа к Интернету. Этим способом сейчас предоставляется доступ к электронной почте и средствам планирования работы, сервисам хранения и обмена файлами, простейшим офисным программам (например, подготовки документов), социальным сетям и т. д.

Подход, при котором в аренду на каких-то условиях предоставляется ресурс, т. е. возможность как услуга, распространяется сейчас не только на готовые сервисы, но и на сами вычислительные ресурсы. Можно арендовать некоторое количество виртуальных компьютеров, определив количество процессорных ядер, оперативной памяти, дискового пространства и каналов связи, развернуть на них свой облачный сервис и наращивать свои требования по мере необходимости.

С технической точки зрения облачный сервис — это целый комплекс, состоящий из взаимодействующих и взаимозаменяемых серверов, соединенных в кластеры, систем хранения данных большой емкости, специально написанного программного обеспечения, высокоскоростных каналов связи. Обслуживается такой сервис (особенно развитый) командами квалифицированных профессионалов, постоянно дорабатывается и развивается. Пользователь, повторимся, из этого не видит ничего — он получает только результат работы, услугу.

Окупаются такие сервисы чаще всего за счет двух источников: за счет модели «Подписка», в которой пользователь оплачивает временную аренду ресурса, или за счет модели «Реклама», в которой пользователю показывается реклама (причем на основе анализа данных и деятельности самого пользователя).

Такой подход распространился и на программное обеспечение: все больше производителей (включая самых крупных) начинают настаивать, чтобы их пользователи переходили от модели покупки программного обеспечения как вещи (коробки) к покупке программы как услуги — на срок, в аренду (SAAS — Software As A Service).

При этом пользователю предоставляются возможность свободного обновления программы, доступ к облачному хранилищу документов, тематическим обсуждениям — но всё это на срок аренды программного обеспечения.

При рассмотрении со школьниками этого материала **будет интересно связать облачные технологии и роботизированные комплексы**, поскольку к автономным роботам облачные сервисы практически не имеют отношения.

Облачные сервисы активно развиваются, в частности, в направлении, получившем название **«Интернет вещей»** (Internet Of Things, IoT). Его базовая концепция — взаимодействие с самыми разными устройствами и вещами (включая самые простые, например, лампы и выключатели) по Сети. Управление такими системами, накопление данных об их функционировании по определению оказываются облачными, потому что на самих устройствах для этого в принципе нет никаких мощностей.

По этому принципу создаются и **системы управления роботизированными производственными комплексами**. При таком подходе, например, программное обеспечение роботов обновляется автоматически, данные об их работе накапливаются в облачной высоконадежной структуре, появляется возможность планировать работу всего комплекса и выдавать задания на основе обработки всего массива данных (например, на основе прогноза заказа товаров заранее распределить их на складе и спланировать график погрузки). Часто можно и самого робота сделать предельно простым — если его среда все равно конструируется и все сложные решения вынесены в облако.

Вполне естественно, что как потребители всевозможных услуг мы должны уметь оценивать плюсы и минусы новых ИТ-сервисов.

Облачные сервисы имеют свои плюсы:

- облачная инфраструктура проектируется как высоконадежная и отказоустойчивая. Отказ диска, потеря и отказ сервера никак не повлияют на работу пользователей — это рядовые рабочие ситуации, которые предусматриваются при разработке;

- ваши данные и услуги доступны вам откуда угодно, причем с минимальными требованиями к устройству (все равно обработка ведется в облаке);
- можно контролировать расходы, а в случае использования минимальных средств и вообще обойтись без них.

Но также **облачные сервисы имеют и существенные минусы:**

- зависимость от канала связи (буквально: авария у провайдера или отказ устройства сети может привести к остановке рабочего процесса);
- зависимость от самого облачного сервиса при полном отсутствии возможности контролировать его работу, получить в случае отказа (а отказы редко, но случаются) какой-то прогноз о сроках восстановления. Причем в некоторых случаях можно потерять свои данные, как правило, без какой-либо адекватной компенсации (формулировка: «Нам жаль, что так получилось, мы готовы дать вам два бесплатных месяца работы» слабо компенсирует потерю готового проекта перед сдачей заказчику);
- в течение длительного срока выплачивая стоимость аренды, пользователь рано или поздно заплатит значительно больше, чем за закупку программного обеспечения, и неизвестно, требуются ли вам как пользователю новые возможности;
- ваши данные доступны как минимум сотрудникам сервиса, а в некоторых случаях и третьим лицам, причем, возможно, по законам не вашей страны (и вы даже не сможете предъявить претензии — вы согласились с этими условиями, начав использовать сервис).

Из достаточно поверхностного взгляда на эти прикладные аспекты использования средств ИТ можно сделать **краткие выводы** о том, что становится актуальным в информационной сфере для всех без исключения:

- Вопросы устройства и работы компьютеров целесообразно изучать в контексте изучения информационных систем, не забывая про информационную безопасность.
- Вопросы искусственного интеллекта становятся неотъемлемой составляющей школьного курса информатики.

Список использованных источников

1. Информационные технологии // Академик. <http://dic.academic.ru/>
2. Калинин И. А., Самылкина Н. Н. Информатика. Углубленный уровень: учебник для 11 класса. М: БИНОМ. Лаборатория знаний, 2014.
3. Примерная основная образовательная программа основного общего образования. Одобрена решением федерального учебно-методического объединения по общему образованию (протокол от 8 апреля 2015 г. № 1/15). http://минобрнауки.рф/документы/938/файл/4587/POOP_OOO_reestr_2015_01.doc

А. А. Салахова,
Московский педагогический государственный университет

ТЕХНИЧЕСКОЕ ТВОРЧЕСТВО И СОРЕВНОВАНИЯ ДЛЯ ФОРМИРОВАНИЯ НОВЫХ КАЧЕСТВ ЛИЧНОСТИ (НА ПРИМЕРЕ РОБОТОТЕХНИЧЕСКИХ СОРЕВНОВАНИЙ)

Аннотация

В статье рассматривается использование робототехнических соревнований между проектными группами в качестве инструмента мотивации обучающихся к техническому творчеству и для формирования у них профессиональных, а также личностных компетенций, необходимых на современном этапе развития общества.

Ключевые слова: проектная деятельность, соревнования, формирование компетенций, техническое творчество, командная работа, открытое инженерное общество.

Контактная информация

Салахова Алена Антоновна, магистрант математического факультета Московского педагогического государственного университета; адрес: 107140, г. Москва, ул. Краснопрудная, д. 14; телефон: (499) 264-25-56; e-mail: aa.salakhova@yandex.ru

A. A. Salakhova,
Moscow State University of Education

TECHNICAL CREATIVITY AND COMPETITIONS FOR FORMING NEW PERSONALITY TRAITS (ON THE EXAMPLE OF ROBOTIC COMPETITIONS)

Abstract

The article discusses the use of robotic competitions among teams and their projects as technical creativity motivation of students and a tool for forming professional and personal traits and competencies which are required by present stage of society development.

Keywords: project activity, competitions, forming competencies, technical creativity, team work, open engineering society.

Сейчас довольно интересное время в образовании и для исследователей, и для практиков. Это период поиска новых активных методов обучения, трансформация традиционных методик обучения, возникновение новых самых разных сочетаний «старого и нового» в образовательной деятельности [6]. Идет переосмысление проектной деятельности в образовании, расширяется применение современных инструментов проектирования (робототехнические платформы и ПО) [1]. Все это меняет наши взгляды на традиционные методы воспитания в образовательной деятельности, среди которых выделим соревнования, поскольку вырос интерес именно к этому методу воспитания в контексте образовательной деятельности. Как и любой другой метод, его надо правильно применять. Например, в сочетании с проектной деятельностью не рекомендуется использовать соревнования внутри проектной группы, это порождает нездоровые конкурентные отношения и вредит как дидактическим целям, так и результату проектной деятельности в целом [5]. В случае реализации практико-ориентированных и творческих проектов как раз рекомендуется проводить конкурс проектов для стимулирования проявления творчества. Конкурс — это и есть соревнование по конкретному виду деятельности.

Есть некоторые требования к организации соревнований, соблюдая которые можно избежать ошибок в применении данного метода воспитания.

Всегда в соревнованиях:

- должен присутствовать творческий компонент помимо исполнительского;
- должна обеспечиваться возможная для всех равнодоступность участия и равновероятность победы;
- для обеспечения равенства возможностей участников оцениваются комплексные показатели;
- при оценивании дополнительно стимулируются взаимопомощь, контактность, значимость результата работы;
- используется прозрачный для участников регламент всех этапов соревнования и критериев оценки.

В соревнованиях по робототехнике совмещаются интеллектуальный, творческий, деятельный и соревновательный (спортивный) компоненты. Такие соревнования не только позволяют демонстрировать результаты, но и формируют необходимые компетенции в разных сферах [3]. Безусловно, робототехнические соревнования являются дополнительным стимулом для технического творчества. Именно соревновательный компонент побуждает учащихся постоянно совершенствовать собственные коды и конструкции роботов.

Робототехнические соревнования в России значительно отличаются от американской или европейской моделей, где их основной задачей является общение технической молодежи, а не победа [1]. В качестве подтверждения этой точки зрения можно привести критерии выбора победителей: в большинстве американских фестивалей победитель определяется без выставления баллов и объявления конкретных преимуществ конструкции или особенностей прохождения трассы. Для российских участников и участников из СНГ крайне важно не только принять непосредственное участие, проверить на прочность свою конструкцию, но также и одержать победу. Оба подхода — это крайности, которых следует избегать.

Для достижения максимального результата с методической точки зрения необходимо выделить следующие **задачи робототехнических соревнований** [4]:

- 1) оптимизация конструкции роботов для наиболее быстрого и качественного прохождения заданий соревнования;
- 2) получение опыта командной работы;
- 3) обмен опытом с другими командами и учителями;
- 4) формирование творческих объединений и сообществ;
- 5) развитие навыка критического мышления, умения принимать решения в стрессовых ситуациях;
- 6) развитие навыка взаимовыручки и поддержки.

Рассмотрим подробнее эти задачи с точки зрения не только формирования новых качеств обучающихся, но и непосредственно самого учителя.

Первый пункт — **оптимизация конструкции роботов для наиболее быстрого и качественного прохождения заданий соревнования** — достаточно прозрачен. Он говорит о необходимости постоянного совершенствования роботов для создания тенденции повышения результатов. Элементы конструкций или даже роботы полностью применяются в следующих сезонах, они не являются разово представленной моделью. Это позволяет эффективно распределять время подготовки, не тратя его на уже готовые удачные решения. Однако и здесь есть дополнительное условие: по-настоящему совершенствование возможно при наличии преемственности в команде. Даже если обучающиеся выступают в индивидуальных соревнованиях, подготовка в стенах школы проходит вместе с другими ребятами. Ученики разных классов под руководством одного педагога могут не только поддержать друг друга, но и действительно помочь. Непосредственное участие других людей в сборке или программировании недопустимо, однако старшие товарищи могут (и должны) поделиться опытом проведения соревнований, объяснить младшим, как происходит организационная часть, какие решения они видели, какие комментарии получили от судей. Кроме того, если принципы работы механизмов и их узлов объясняет товарищ, ребенок усваивает информацию быстрее, чем если эту же тему ему расскажет взрослый. Для поддержания преемственности можно устраивать общие встречи, товарищеские соревнования между разными возрастными категориями, вести общий для школы альбом соревнований. Например, все команды FIRST Tech Challenge, в которых соблюдается принцип преемственности, вышли в финал FISRT Tech Challenge Russia 2017. Преподаватели также отметили, что многие выпускники этих команд продолжили свой путь в робототехнике, включая основание собственных робототехнических образовательных проектов. В командных соревнованиях соблюдение принципа преемственности позволяет обучающемуся не только попробовать разные роли, но и поэтапно изменять степень ответственности. Так, в первый год участия происходит погружение в атмосферу соревнований, знакомство с фестивалями, регламентом, расписанием, другими требованиями. Второй год — это непосредственное участие в команде, выполнение роли, изучение конструкций, программирования, обсуждение идей. На третий год и далее возможно принятие позиции капитана команды, управление командой, окончательное утверждение решений остальных участников и помощь учителю. Одна школьная команда может участвовать в различных соревнованиях, в том числе по уровням сложности требований регламентов, и тогда дополнительным инструментом для поддержания преемственности может стать обучение старшими

ребятами младших по использованию дополнительных возможностей робототехнических наборов [2]. Именно такая работа с командой помогает развитию у детей критического мышления, умения принимать решения в стрессовой ситуации, т. е. в ситуации соревнований.

Таким образом, мы плавно перешли ко второму пункту — **получение опыта командной работы**. Команда — это не только объединение по интересам, но и сплоченная бригада, в которой каждый четко знает свою роль и обязанности. Помимо участия в команде ребята активно **участвуют в творческих объединениях и сообществах** и организуют их, т. е. формируют активную социальную позицию, в том числе проводя различные мастер-классы и выступления как для детей младшего возраста, так и ровесников. Роли в команде могут быть мобильны в различные сезоны фестивалей или при подготовке к разным мероприятиям, однако менять роль внутри одного сезона не рекомендуется. Перед началом соревнований команда распределяет роли на собрании и ставит перед каждым членом цели и задачи. Список задач может меняться в течение года в пределах примерно 30–35 %, это нормально и связано с техническими или программными сложностями, обстоятельствами, которые могли возникнуть не по вине обучающихся. Задачи, цели и роли каждого участника команды в соревнованиях и прописываются в документе, который для некоторых соревнований приобретает формат инженерной книги [4]. Каждый участник должен осознавать свои обязанности и планировать деятельность. Общим контролем занимается менеджер проекта (под проектом здесь понимается сезон конкретного соревнования) — это капитан команды или избранный всеми ответственный, обычно старший участник. Он контролирует выполнение распределенных в течение года задач и достижение целей, помогает выяснить причины, если задачи не были достигнуты, принимает решение о дополнительных встречах и тренировках. Учитель не должен забирать эту роль у своих подопечных, однако он и не должен полностью отказываться от функций управления: педагог ненавязчиво дублирует контроль и подводит ответственного к решениям в этой сфере, если возникают трудности. Учитель также полностью забирает на себя подготовку трансфера, проживания, питания и других сопутствующих соревнованиям расходов и дел, обеспечивает безопасность ребят. Детей ничто не должно отвлекать от технического творчества, особенно усталость или чувство голода. Кроме того, общение команды не должно быть ограничено очными встречами в школе и на площадке: переписки, общий блог, сервисы для командной работы — единое электронное пространство для взаимодействия, выполнения задач очень важно. Умение пользоваться подобными сервисами высоко ценится в профессиональной деятельности, а также при дальнейшем обучении в вузе. Умение работать внутри команды — это не только наличие атмосферы взаимопонимания, но и способность без заминки решать задачи, функционируя, как множество узлов единого механизма.

Взаимодействие должно быть также с другими командами. На региональных или всероссийских соревнованиях возможно завязывание новых связей, дружбы между представителями разных регионов. Если ребята находят собеседников довольно быстро, то иначе обстоит дело с учителями. Педагоги также имеют возможность взаимодействия, обмена опытом и получения дополнительных контактов для обеспе-

чения мобильности команды на будущих соревнованиях. Основная ошибка, которую допускают команды и педагоги, заключается в том, что межкомандное взаимодействие ограничивается площадкой фестивалей. Это неправильный подход. Современные технологии позволяют поддерживать связь в течение всего года, используя различные творческие объединения и сообщества. Не стоит бояться обсуждать решения: в конечном итоге конструкции будут различны, даже если юные конструкторы отталкивались от одной идеи. Общение между командами помогает развивать навыки коммуникации, способствует формированию в будущем открытого инженерного сообщества. Кроме того, ведение блогов, видеоканалов в YouTube или интернет-журналов делает занятия робототехникой разнообразней, влияет на личностные качества и развивает навыки, которые необходимы в будущей профессиональной деятельности не только в ИТ-сфере. Особое внимание нужно уделить международному общению: живое взаимодействие развивает навыки коммуникации на иностранных языках, делает речь свободной и естественной, а также воспитывает поликультурность, толерантность, убирает сложности взаимодействия в обществе представителей различных национальностей, носителей различных языков и традиций [4]. Дети, принимающие участие в международных соревнованиях или хотя бы поддерживающие общение с иностранными командами через мессенджеры, такие как Skype, WhatsApp, Viber, с помощью Twitter, Instagram и через другие социальные сети и инструменты, готовы к профессиональному взаимодействию с любыми членами команды, даже если та сформирована только что.

Что касается пункта о **развитии навыка взаимовыручки и поддержки**, то здесь прежде всего нужно отметить, что нельзя запрещать помогать другим. Во-первых, это не приведет к поражению своей команды, наоборот, сделает игру (соревнование) интересней. Взаимопомощь — это важное свойство открытого инженерного общества. Во-вторых, возникает необходимость быть постоянно готовыми к решению возникших проблем у других участников, в том числе уметь быстро найти

и устранить поломку в совершенно незнакомых конструкциях, что предполагает формирование и развитие умения быстро понимать ход чужих мыслей и инженерных идей. Таким образом, взаимовыручка и помощь другим командам во время соревнований (при тестовых заездах и подготовке), характерная американской модели, развивает **навык критического мышления и умения принимать решения в стрессовых ситуациях**. В-третьих, взаимовыручка развивает положительные личностные качества.

Соревнований и олимпиад становится много. Это скорее положительная тенденция — у обучающихся есть выбор по интересам и возможностям. Для учителя становится непросто отслеживать сроки и требования разных соревнований. Поэтому, конечно, стоит выбрать наиболее интересные именно вашим ученикам и вам соревнования для систематической подготовки к ним.

Список использованных источников

1. Кроули Э. Ф., Малмквис Й., Остлунд С., Бродер Д. Р., Эдстрем К. Переосмысление инженерного образования. Подход CDIO / пер. с англ. С. Рыбушкиной; под науч. ред. А. Чучалина. М.: Изд. дом Высшей школы экономики, 2015. (Библиотека журнала «Вопросы образования».)
2. Салахова А. А. Введение в текстовые языки программирования средствами робототехнического набора LEGO Mindstorms EV3 // Мониторинг образовательной робототехники и ИТ-образования города Москвы. М.: Изд. центр АНО «АИР», 2017.
3. Салахова А. А. Образовательная робототехника как средство реализации проектной деятельности школьников. Учебные пособия // Инновационные технологии в науке, технике, образовании: материалы Всероссийской научно-практической конференции (17 марта 2017 года, г. Саратов). Саратов: Научная книга, 2017.
4. Салахова А. А. FIRST Tech Challenge. Путь к победе. М.: Лаборатория знаний, 2017. (Серия «Фестиваль робототехники».)
5. Таранова В. В., Самылкина Н. Н. Робототехника в школе. Методика, программы, проекты. М.: Лаборатория знаний, 2017.
6. Martinez J. E. The Search for Method in STEAM Education. London: Palgrave Macmillan, 2017.

ПРАВИЛА ДЛЯ АВТОРОВ

Уважаемые коллеги!

Статьи для публикации в журналах «Информатика и образование» и «Информатика в школе» должны отправляться в редакцию **только через электронную форму на сайте ИНФО (раздел «Авторам → Отправка статьи»):**

<http://infojournal.ru/authors/send-article/>

Обращаем ваше внимание, что для отправки статьи необходимо предварительно зарегистрироваться на сайте ИНФО (или авторизоваться — для зарегистрированных пользователей).

С требованиями к оформлению представляемых для публикации материалов можно ознакомиться на сайте ИНФО в разделе **«Авторам»:**

<http://infojournal.ru/authors/>

Дополнительную информацию можно получить в разделе **«Авторам → Часто задаваемые вопросы»:**

<http://infojournal.ru/authors/faq/>

а также в редакции ИНФО:

e-mail: readinfo@infojournal.ru

телефон: (495) 140-19-86

В. А. Бобров,
Московский педагогический государственный университет

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ НА ОПРЕДЕЛЕНИЕ СООТВЕТСТВИЙ

Аннотация

В статье рассматриваются различные подходы к решению задач на определение соответствий, проводится анализ этих подходов, приводятся примеры задач, решаемых с помощью этих подходов, дается оценка необходимости решения задач на определение соответствий на школьных уроках информатики.

Ключевые слова: школьная информатика, методы решения задач, задачи на определение соответствий.

Контактная информация

Бобров Владимир Андреевич, студент 4-го курса математического факультета Московского педагогического государственного университета; *адрес:* 107140, г. Москва, ул. Краснопрудная, д. 14; *телефон:* (499) 264-25-56; *e-mail:* Bobrov1505@yandex.ru

V. A. Bobrov,
Moscow State University of Education

METHODS OF SOLVING COMPLIANCE PROBLEMS

Abstract

The article describes the approaches to the solution of compliance problems, the analysis of these approaches, contains examples of relevant tasks and review of the need to solving compliance problems on school informatics lessons.

Keywords: school informatics, problem-solving methods, compliance problems.

При рассмотрении темы «Информация и информационные процессы», а именно процесса обработки информации ученикам основной школы зачастую предлагаются задачи на определение (установление) соответствий [1, 2, 3]. Встречаются такого рода задачи и в материалах Международного конкурса по информатике «Бобер» (<http://bebras.ru>). Несмотря на то что решение таких задач не требует никаких специальных знаний, а сами задачи больше напоминают загадки на сообразительность, для успешной работы с ними необходимо выработать специальный подход.

Под **задачами на определение соответствий** будем понимать текстовую задачу, в которой:

- 1) речь идет об объектах нескольких — не менее двух — множеств (имен, фамилий, городов, видов спорта и т. д.), состоящих из одинакового количества элементов; совокупность всех объектов каким-то образом разбита на группы, в каждой из которых находится ровно по одному элементу из каждого множества;
- 2) информация представлена в виде текста или набора утверждений и сводится к перечню тезисов о том, что какие-то два конкретных объекта (из разных множеств) находятся или не находятся в одной и той же группе. Все эти данные будут считаться при решении заведомо верными, их должно быть достаточно, чтобы однозначно определить разбиение объектов на группы без возникновения противоречий.

Обычно в таких задачах имена объектов начинаются с разных букв либо элементы одного из множеств можно обозначить цифрами. Это позволяет упростить обозначения объектов в процессе решения задачи, сократив их имена до одного символа.

Рассмотрим пример такой задачи.

Задача 1.

Антон, Борис, Влад и Георгий — ученики средней школы. Первого сентября они пришли на свой первый урок информатики. Им так понравился новый предмет, что они сразу же захотели записаться в школьный кружок по программированию. «Дубин, у тебя есть замаска?» — спросил у соседа по парте Влад. «Нет, попроси у Ясенева!» — ответил ему Антон. «Липин, передай мне, пожалуйста, замаску», — попросил Влад другого своего одноклассника.

Когда Иван Иванович, их учитель информатики, после урока посмотрел на список записавшихся, то он увидел, что ученики написали только свои фамилии: Дубин, Ельнев, Липин и Яснев. У него не было под рукой списка класса, но он хорошо помнил имена мальчиков и весь их диалог, а еще был уверен, что фамилия Бориса — не Липин. Подумав несколько секунд, он легко смог подписать к каждой фамилии имя.

Как после этого стал выглядеть список?

Решение.

Для решения задачи, в которой необходимо сопоставить элементы двух множеств, классическим приемом является составление таблицы [2]. В названиях ее строк выписываются элементы одного множества, в названиях столбцов — элементы второго. На пересечениях строк или столбцов ставятся знаки «плюс» или «минус». «Плюс» означает, что элементы точно находятся в одной группе, а «минус» — что они точно находятся в разных группах. Составим такую таблицу для нашей задачи.

Из разговора учеников можно установить следующее:

- 1) фамилия Влада — не Дубин;
- 2) фамилия Антона — не Яснев;
- 3) фамилия Антона — Дубин;
- 4) фамилия Влада — не Яснев;
- 5) фамилия Влада — не Липин.

Еще одно утверждение следует из того, что известно самому учителю:

- 6) фамилия Бориса — не Липин.

Конечно, п. 1 и 2 — следствия п. 3, т. е. информация п. 1 и 2 является избыточной. Но при решении с помощью табличного подхода ученика не напугает эта проблема, поскольку алгоритм действий при этом никак не изменяется.

Итак, составим таблицу, используя утверждения 1–6:

	Дубин	Ельнев	Липин	Ясенев
Антон	+			–
Борис			–	
Влад	–		–	–
Георгий				

После того как в таблицу занесены начальные данные, необходимо заполнить ее, используя два правила:

- 1) если в какой-то ячейке находится «плюс», то необходимо поставить «минусы» во все ячейки того же столбца и той же строки;
- 2) если во всех ячейках какого-то столбца или строки, кроме одной, стоят «минусы», то необходимо поставить «плюс» в оставшуюся свободную ячейку.

В случае каждой конкретной задачи рекомендуется обсудить с учениками смысл расставляемых знаков «плюс» и «минус». Например, если фамилия Антона Дубин, то он точно не Ельнев, не Липин, не Ясенев; если Дубин Антон, то он точно не Борис, не Влад и не Георгий.

Продолжая применять правила 1 и 2, заполним таблицу до конца:

	Дубин	Ельнев	Липин	Ясенев
Антон	+	–	–	–
Борис	–	–	–	+
Влад	–	+	–	–
Георгий	–	–	+	–

Задача решена. Теперь остается только «прочитать» ответ по таблице:

- Антон — Дубин,
- Борис — Ясенев,
- Влад — Ельнев,
- Георгий — Липин.

Для того чтобы убедиться в правильности полученного ответа, рекомендуется еще раз прочесть условие задачи и убедиться, что полученные соответствия имен и фамилий ему не противоречат.

Рассмотрим более сложную задачу, в которой фигурируют не два, а три множества объектов.

Задача 2.

Антон, Денис и Юра вошли в тройку призеров школьного соревнования по многоборью (т. е. заняли первое, второе и третье места). Оказалось, что каждый из них занимается одним из трех видов спорта: футбол, хоккей (с шайбой), баскетбол. Необходимо определить, какое место занял каждый ученик и каким видом спорта он занимается, если известно, что:

- 1) мальчик, занявший третье место, увлекается не хоккеем;
- 2) Антон занял не третье место;

- 3) ученик, ставший на соревнованиях вторым, занимается баскетболом;
- 4) Юра занял не второе место;
- 5) Юра никогда не играл в футбол.

Решение.

Нарисовать таблицу, аналогичную рассмотренной в задаче 1, не получается: в нашем случае такая таблица должна быть трехмерной (похожей на кубик Рубика), что изобразить на плоском листе бумаги весьма затруднительно.

Первый способ решения такой задачи — составить три таблицы вместо одной. При этом правила для составления этих таблиц аналогичны рассмотренным при решении задачи 1. Каждая таблица будет отражать связи объектов в одной из трех пар множеств, и в данном случае (если заполнить их известными данными) они будут выглядеть так:

	Ф	Х	Б
Антон			
Юра	–		
Денис			

	1	2	3
Антон			–
Юра		–	
Денис			

	Ф	Х	Б
1			
2			+
3		–	

Правила заполнения каждой из таблиц остаются прежними, но теперь при решении необходимо также переносить данные из одной таблицы в другую для каждого известного соответствия каких-либо двух объектов (т. е. при наличии в одной из таблиц плюса).

Например, поскольку мальчик, занявший второе место, увлекается баскетболом, то все, что известно про серебряного призера, верно и для баскетболиста (и наоборот). В частности, раз второе место занял не Юра, то и в баскетбол играет не он.

Именно в этой новой необходимости синхронизации таблиц и заключается минус такого способа: проверять, вся ли информация из одной таблицы перенесена в другую, достаточно сложно, а запутаться при проверке очень легко.

Рассмотрим **второй способ** решения задачи.

Оставим из трех таблиц только две, причем объединим их при этом в одну [2]. Для этого нужно выбрать одно множество объектов как основное, а два других тогда окажутся связанными между собой через него. Интуитивно проще в качестве основного множества выбрать то, которое естественным образом обозначает основных действующих лиц задачи. Элементы основного множества при составлении выписываются в централь-

ном столбце, а элементы двух оставшихся — в названиях столбцов. Правила работы с каждой половиной таблицы остаются все теми же.

В данном случае в качестве основного множества мы выберем имена спортсменов. Выглядеть наша таблица будет следующим образом:

1	2	3		Футбол	Хоккей	Баскетбол
		—	Антон			
	—		Юра	—		
			Денис			

Минус этого способа виден сразу — вне зависимости от выбора основного множества занести всю информацию из утверждений в таблицу, скорее всего, не получится (для данной задачи это тоже верно). Это означает, что для решения задачи будет необходимо работать не только с самой таблицей, но и с исходным перечнем утверждений.

Например, чтобы сделать все тот же вывод, что Юра не занимается баскетболом, необходимо не только посмотреть на таблицу, но и прочитать утверждение 3.

Таким образом, решение задачи данным способом сводится не только к применению правил, рассмотренных в задаче 1, но и к поиску и использованию подходящих утверждений из условия, что позволило бы внести в таблицу недостающую информацию.

Рассмотрим *третий способ*, позволяющий решать задачу, не возвращаясь к условию после внесения начальных данных в таблицу.

Составим таблицу по следующему принципу: в названия столбцов вынесем элементы одного множества, в названия строк — элементы другого, а элементы третьего обозначим одним символом и поместим внутри каждой из ячеек. Элементы третьего множества внутри ячейки можно размещать любым способом, например, в линию или по вершинам правильного многоугольника (с количеством вершин, равным количеству элементов множества).

	Футбол	Хоккей	Баскетбол
Антон	1 2 3	1 2 3	1 2 3
Юра	1 2 3	1 2 3	1 2 3
Денис	1 2 3	1 2 3	1 2 3

Обозначать информацию в таблице будем следующим образом:

- 1) если какая-то из групп полностью определена (т. е. мы узнали, каким видом спорта увлекается ученик и какое он занял место), то нужно заключить в прямоугольную рамочку элемент третьего множества, находящийся в ячейке, расположенной на пересечении строки и столбца, соответствующих элементам первого и второго множеств. Например, если будет известно, что Антон играет в хоккей и занял первое место, в рамочку нужно заключить единицу в клетке первой строки второго столбца;
- 2) если точно известно, что какой-то набор элементов не может быть в одной группе (назовем такой набор *несовместным*), то нужно вычеркнуть соответствующий этому набору элемент.

Тогда в конце решения в ответе окажется таблица, в которой:

- ровно один прямоугольник в каждой строке;
- ровно один прямоугольник в каждом столбце;
- каждый из элементов третьего множества заключен в прямоугольник ровно в одной из ячеек.

Процесс решения состоит в вычеркивании несовместных наборов. На примере данной задачи необходимо следовать следующим шести правилам (в других задачах эти шесть правил формулируются аналогично с точностью до объектов множеств):

- 1) если какой-то ученик не играет в какой-то из видов спорта, вычеркиваются все цифры в ячейке на пересечении строки этого ученика и столбца этого вида спорта;
- 2) если какой-то ученик играет в определенный вид спорта, вычеркиваются все цифры в строке ученика и в столбце вида спорта, за исключением находящихся в ячейке на их пересечении;
- 3) если известно место, которое не занял ученик, вычеркиваются все цифры, соответствующие этому месту, в строке ученика;
- 4) если известно место, которое ученик занял, вычеркиваются все цифры других мест в строке ученика и все цифры, соответствующие этому месту, в строках других учеников;
- 5) если известно, какое место не занял мальчик, занимающийся в определенной спортивной секции, вычеркиваются все цифры, соответствующие этому месту, в столбце вида спорта;
- 6) если известно, какое место занял мальчик, занимающийся в определенной спортивной секции, вычеркиваются все цифры других мест в столбце этого вида спорта и все цифры, соответствующие этому месту, в столбцах других видов спорта.

К утверждению 1 из условия применим правило 5, к утверждению 2 — правило 3, к утверждению 3 — правило 6, к утверждению 4 — снова правило 3, а к утверждению 5 — правило 1.

После учета утверждений 1–5 таблица примет следующий вид:

	Футбол	Хоккей	Баскетбол
Антон	1 2 3	1 2 3	+ 2 3
Юра	+ 2 3	1 2 3	+ 2 3
Денис	1 2 3	1 2 3	+ 2 3

В строке Юры осталась всего лишь одна незачеркнутая цифра, значит, можно сделать вывод о том, что Юра играет в хоккей и занял первое место. Это позволяет вычеркнуть из таблицы все единицы и все цифры второго столбца:

	Футбол	Хоккей	Баскетбол
Антон	+ 2 3	+ 2 3	+ 2 3
Юра	+ 2 3	1 2 3	+ 2 3
Денис	+ 2 3	+ 2 3	+ 2 3

В таблице осталась всего лишь одна тройка. Из этого можно сделать вывод, что Денис занял третье место и играет в футбол. Вычеркнув все другие цифры из строки Дениса и столбца футбола (правда, столбец в данном случае уже и так «пуст»), мы приходим к финальной форме таблицы:

	Футбол	Хоккей	Баскетбол
Антон	1 2 3	1 2 3	1 2 3
Юра	1 2 3	1 2 3	1 2 3
Денис	1 2 3	1 2 3	1 2 3

Получаем *ответ*: Антон занял второе место и играет в баскетбол, Юра занял первое место и играет в хоккей, Денис занял третье место и играет в футбол.

Примечание. Иногда (при решении намного более сложных задач того же типа) для получения следующего шага следует выводить и затем сразу использовать новые утверждения. Например, если в строке остались только одинаковые цифры, то можно сделать вывод, что ученик, чье имя находится в этой строке, занял место, соответствующее этим цифрам. Это, в свою очередь, позволит по правилу 4 вычеркнуть все цифры, сопоставленные данному месту, в других строках.

В качестве еще одного примера рассмотрим задачу из учебника информатики для девятого класса [2].

Задача 3.

Три подружки — Аня, Света и Настя — купили различные молочные коктейли в белом, голубом и зеленом стаканчиках. Ане достался не белый стаканчик, а Свете — не голубой. В белом стаканчике не банановый коктейль. В голубой стаканчик налит ванильный коктейль. Света не любит клубничный коктейль. Требуется выяснить, какой коктейль и в каком стаканчике купила каждая из девочек.

Решение.

В учебнике [2] описывается вариант решения данной задачи с помощью двух таблиц. Мы же решим эту задачу последним из описанных в задаче 2 способов, используя всего одну таблицу.

Изначально таблица будет иметь такой вид:

	Банановый	Ванильный	Клубничный
Аня	Б Г З	Б Г З	Б Г З
Света	Б Г З	Б Г З	Б Г З
Настя	Б Г З	Б Г З	Б Г З

Внесем в нее данные из условия:

	Банановый	Ванильный	Клубничный
Аня	Б Г З	Б Г З	Б Г З
Света	Б Г З	Б Г З	Б Г З
Настя	Б Г З	Б Г З	Б Г З

Сразу можно сделать вывод, что у Светы зеленый стаканчик и в него налит банановый коктейль. После этого можно оставить букву «З» только в ячейке на пересечении строки «Света» и столбца «Банановый», а также вычеркнуть все буквы из других ячеек этого столбца. Таблица примет вид:

	Банановый	Ванильный	Клубничный
Аня	Б Г З	Б Г З	Б Г З
Света	Б Г З	Б Г З	Б Г З
Настя	Б Г З	Б Г З	Б Г З

Далее можно сделать вывод о том, что в белый стаканчик налит клубничный коктейль и купила его Настя, ведь буква «Б» осталась только в одной клетке. Теперь можно вычеркнуть все другие буквы в последней строке и выделить в рамку оставшуюся букву «Г»:

	Банановый	Ванильный	Клубничный
Аня	Б Г З	Б Г З	Б Г З
Света	Б Г З	Б Г З	Б Г З
Настя	Б Г З	Б Г З	Б Г З

Ответ. Аня купила ванильный коктейль в голубом стаканчике, Света купила банановый коктейль в зеленом стаканчике, Настя купила клубничный коктейль в белом стаканчике.

Приведем еще четыре задачи для самостоятельного решения.

Задача 4.

Андрей, Боря и Витя пригласили на весенний бал девочек из своего класса — Катю, Лену и Машу. При этом каждый из них преподнес своей избраннице один из трех букетов — из роз, из сирени и из тюльпанов, причем букеты не повторялись. Достоверно известно:

- 1) Андрей пригласил именно Машу;
- 2) Борис не приглашал Лену и не покупал букета из роз;
- 3) Витя не покупал букет из цветов сирени;
- 4) Катя получила букет не из цветов сирени.

Кто из мальчиков пригласил какую девочку и подарил какой букет?

(Андрей пригласил Машу, подарив букет из сирени; Боря пригласил Катю, подарив букет из тюльпанов; Витя пригласил Лену, преподнес ей букет из роз. В ячейках рекомендуется указывать тип букета.)

Задача 5.

Три брата-царевича разных возрастов — Иван, Василий и Дмитрий — прискакали к развилке дорог: одна вела налево, вторая — направо, а третья — прямо. «Куда путь держать думаете?» — обратился Иван к своим младшим братьям. «Не поеду я направо!» — заявил Василий, не самый молодой из троих. «Если так, не быть тому, чтобы ехал я налево!» — молвил ему в ответ Дмитрий. И каждый из братьев двинулся по своей дороге.

Определите (по именам), кто из братьев отправился по какой дороге, а также, кто из них был старшим, кто

средним, а кто младшим, если младший брат поехал не направо.

(Иван, старший брат, поехал направо; Василий, средний брат, поехал налево; Дмитрий, младший брат, поехал прямо. В ячейках рекомендуется указывать первую букву имени, так как только имена начинаются с разных букв.)

Задача 6.

На одном сказочном острове живут два народа: правдцы и хитрецы. Правдцы говорят только правду, а хитрецы могут говорить и правду, и ложь (по своему желанию). И вот однажды на этом острове произошло ограбление банка, из него по очереди вынесли три сундука: маленький, средний и большой. В одном из них была медь, в другом — серебро, а в третьем — золото. Сотрудник банка, один из жителей острова, утверждает, что видел ограбление и полностью уверен в своих словах. Его показания выглядели так:

«Я точно помню, что первым грабители вынесли не большой и не средний сундук и лежала в нем точно не медь. Кстати, в маленьком сундуке было не золото. Ах да, чуть не забыл: серебро вынесли не в первом сундуке!»

Стоит ли следователям безоговорочно верить каждому сказанному им слову?

(Житель, дававший показания, не может быть правдцом, поскольку в его словах есть противоречие. Значит, каждому его слову верить не надо. Противоречие будет явно заметно при решении третьим способом. Если обозначить сундуки (в порядке выноса) цифрами от 1 до 3, то после занесения в таблицу начальных условий в ней либо останется пустой строка или пустым столбец, соответствующий цифре 1, либо не останется цифр 1 в ячейках (в зависимости от того, где будут размещены цифры: в названиях столбцов, в названиях строк или в ячейках). Это означает, что нельзя совместить цифры, размеры и содержимое всех сундуков так, чтобы все условия в показаниях выполнялись.)

Задача 7.

В финал соревнований по бегу вышли четыре спортсмена. Они представляли Россию, Перу, Словакию и Турцию. У каждого была майка своего цвета: белая, красная, синяя или зеленая. Все финалисты успешно добрались до финиша, тройка победителей определилась без деления мест.

Известно, что:

- 1) российский спортсмен выступал в синей майке;
- 2) спортсмен в зеленой майке завершил дистанцию последним;
- 3) турецкий спортсмен выступал не в зеленой майке, а словацкий — не в красной;
- 4) словацкий спортсмен не оказался последним;
- 5) турецкий спортсмен проиграл несколько секунд победителю;
- 6) серебро досталось спортсмену, выступавшему не в синей и не в красной майке.

Определите место и майку спортсмена из каждой страны.

(Первое место занял спортсмен из России, выступавший в синей майке; второе место занял спортсмен из Словакии, выступавший в белой майке; третье место занял спортсмен из Турции, выступавший в красной майке; четвертое место занял спортсмен из Перу, выступавший в зеленой майке. В ячейках таблицы рекомендуется указывать занятое место.)

* * *

В заключение заметим, что способ решения задачи 1 и третий способ решения задачи 2 описывают универсальные подходы к решению задач на соответствие и позволяют свести текстовое условие к табличной форме, в рамках которой возможно решить задачу.

Задачи на определение соответствий помогают развить математическое мышление у школьников, отработать навык построения модели на основе текста условия, демонстрируют важность умения выполнять алгоритм, а также позволяют это умение отработать. Простая, без использования большого количества научных терминов формулировка положительно сказывается на мотивации школьников при решении задач такого типа. С задачами на определение соответствий школьник может встретиться и при участии в олимпиадах. Все это позволяет сделать вывод о безусловной пользе отработки навыка решения подобных задач на уроках информатики.

Список использованных источников

1. Босова Л. Л., Босова А. Ю. Информатика: учебник для 7 класса. М.: БИНОМ. Лаборатория знаний, 2013.
2. Босова Л. Л., Босова А. Ю. Информатика: учебник для 9 класса. М.: БИНОМ. Лаборатория знаний, 2013.
3. Босова Л. Л., Босова А. Ю., Коломенская Ю. Г. Занимательные задачи по информатике. 5-е изд. М.: БИНОМ. Лаборатория знаний, 2014.

Н. А. Арнаутова,
Московский педагогический государственный университет

ПРИМЕНЕНИЕ САМОДЕЛЬНОЙ МУЛЬТИПЛИКАЦИИ НА РАЗНЫХ ЭТАПАХ УРОКА

Аннотация

Статья посвящена проблеме эффективного соединения ИКТ-компетентности учителя-предметника с методикой преподавания его предмета. Рассматривается применение самодельной мультипликации в качестве продуктивного метода обучения. Даются рекомендации по созданию эффективных формулировок заданий в зависимости от этапа урока.

Ключевые слова: мультимедийные технологии, самодельная мультипликация, методика преподавания, продуктивные методы обучения.

Контактная информация

Арнаутова Наталья Александровна,
ст. преподаватель кафедры информационных технологий в образовании Московского педагогического государственного университета;
адрес: 119571, г. Москва, пр-т Вернадского, д. 88, каб. 732; телефон: (495) 438-11-53; e-mail: arnautova@mail.ru

N. A. Arnautova,
Moscow State University of Education

THE APPLICATION OF DIY ANIMATION FOR DIFFERENT STAGES OF LESSON

Abstract

The article is devoted to the problem of effective connection of the ICT competence of the teacher of any school subject with the teaching methods of its subject. The application of DIY animation as a productive method of learning is considered. Recommendations are given to create effective formulations of tasks according the stage of the lesson.

Keywords: multimedia technologies, DIY animation, methodics of teaching, productive teaching methods.

Уровень современных требований к ИКТ-компетентности учителя-предметника среднего и старшего звена или учителя начальной школы таков, что подразумевает его полную самостоятельность в организации мультимедийных уроков по своему профилю (или в применении мультимедийных технологий как элементов урока). Современный учитель должен не только сам, без помощи коллеги-информатика спланировать такие занятия и провести их, но и быть в состоянии объяснить особенности применения соответствующего цифрового оборудования и компьютерных технологий своим ученикам. Однако следует помнить, что, обладая требуемым уровнем ИКТ-компетентности, надо еще уметь правильно его соединить с методикой преподавания своего предмета.

Самодельная мультипликация — в технологии покадровой фотосъемки последовательных фаз движения или в виде получения мультипликата как набора изображений последовательных фаз в графических редакторах на компьютере — является достаточно гибким мультимедийным инструментом, способным повысить эффективность образовательного пространства.

Попробуем дать некоторые выработанные достаточно большим учительским опытом **рекомендации по применению самодельной дидактической мультипликации на уроках любого профиля.**

Ученики относительно изучаемого материала могут находиться в трех основных позициях:

- 1) материал изучен, и требуется проверка основательности понимания;
- 2) материал еще не изучен, и только предстоит с ним знакомиться;
- 3) материал недавно разобран и находится в процессе освоения и усвоения.

Задания, использующие дидактическую мультипликацию, обязательно должны учитывать особенности этих трех этапов и плюс еще некоторые условия, о которых скажем позже. Главное, в каждом случае это должно быть задание не репродуктивное, а продуктивное, не описание или простое воспроизведение, а всегда плод раздумий, поиска решений.

Заметим, что речь здесь идет о заданиях, которые выполняются индивидуально или в микрогруппах учениками именно на уроке и могут, по усмотрению учителя и исходя из складывающейся образовательной ситуации, занимать как часть урока, так и весь урок.

1. Материал изучен, и требуется проверка основательности понимания.

Итак, предположим, что материал уже изучен ранее и, как ожидается, достаточно хорошо усвоен. Как сформулировать вопрос, чтобы получение ответа на него в виде мультфильма было максимально полезно ученику и рассказывало учителю о степени усвоенности знаний? В данном случае рекомендуется в вопросе ставить задачу, решение которой можно получить, только опираясь на изученный материал.

Разберем примеры.

Допустим, ранее на уроках (биология, VI класс) была изучена тема «Клеточное строение листа». Задание на создание мультфильма в таком случае может быть таким:

«Рассматривая ситуацию на клеточном уровне, в виде мультфильма покажите, что произошло бы с растением, листья которого утратили клетки кожицы».

Варианты этого задания получаются, если вместо клеток кожицы учитель вводит «утрату» клеток других тканей, составляющих лист растения.

Пример из области физики. Пусть ранее учениками VII класса была изучена тема «Водяной насос». Тогда задание может звучать так:

«В виде мультфильма покажите ситуацию, когда у водяного насоса в ходе работы случилась поломка одной из его деталей».

Варианты задания можно получить, если вместо водяного насоса выбрать другие приборы и измерительные инструменты, рассматриваемые в курсе физики.

2. Материал еще не изучен, и только предстоит с ним ознакомиться.

Теперь допустим, что материал еще не изучен и мы решаем, как познакомить ученика с ним. В таком случае можно предложить текст с изложенным материалом и попросить снять мультфильм по этому тексту, как по сценарию. В такой ситуации это уже не будет простой репродукцией, так как потребует сложного мыслительного процесса по получению собственной, выраженной средствами мультфильма, интерпретации материала.

Вернемся к темам предыдущих примеров.

Пусть тему «Клеточное строение листа» только предстоит изучить. Тогда задание формулируется так:

«Внимательно разобрав текст параграфа и рассмотрев иллюстрации,ними мультфильм, в котором листок растения правильным образом складывается из всех видов клеток, словно из деталей конструктора».

На уроке физики задание по теме «Водяной насос» будет таким:

«Прочитав в учебнике параграф об устройстве и работе водяного насоса,ними мультфильм, показывающий во всех деталях, как эта работа происходит».

Еще один ход, которым можно пользоваться в момент знакомства с новым, это предложение сравнения, сопоставления нового материала с ранее изученным. Например, найти сходства и различия и выразить языком мультфильма. То есть, обращаясь к приведенным выше темам, при первичном изучении материала можно сформулировать задание по биологии так:

«После прочтения материала параграфа о клеточном строении листаними мультфильм, показывающий, чем клеточное строение листа и корня похожи и чем отличаются».

Это, конечно, при условии, что клеточное строение корня уже было изучено.

3. Материал недавно разобран и находится в процессе освоения и усвоения.

Пусть материал уже был разобран каким-то образом ранее и ученик находится на стадии усвоения, закрепления материала. В таком случае будет правильно формулировать такие задания, в которых предлагается обратить внимание на какие-то особенности в теме, о которых хорошо знает учитель и которые могут ускользнуть от внимания ученика. К мультипликации тогда правильно предлагать какие-то подробности, частности, уникальности, чтобы погрузить ученика в материал как можно более глубоко.

В частности, задание на уроке физики в VII классе может быть таким:

«В мультфильме покажи, каким образом работа водяного насоса связана с атмосферным давлением».

Важно понимать, что должен соблюдаться баланс трудоемкостей. То есть трудоемкость съемки мультфильма не должна превышать трудоемкость поиска ответа на заданный вопрос, она не должна становиться основной трудоемкостью. Другими словами, если ответить на поставленный вопрос можно быстро и легко парой предложений, то предлагать делать ответом на такой вопрос

мультипликационный ролик не надо категорически. Нарушение такого баланса приведет к раздражению и появлению большого сопротивления у ученика, что не будет способствовать успеху в изучении материала и уважению к самодельной мультипликации как инструменту получения знаний. И наоборот, если мы хотим избежать легковесных, скоропалительных суждений при поиске ответа, то есть хотим включить обдумывание, то съемка становится нашим помощником.

Например, на уроках физики в VIII классе была изучена тема «Строение атома и атомного ядра», и учитель собирается проверить знания по ней. Если он формулирует задание для съемки так: «Сними мультфильм про строение атома», то он не учитывает, что перечислить названия элементарных частиц, входящих в состав атома, куда проще устно или письменно в виде перечисления названий и количества частиц для конкретно взятых атомов. Тема в школьном курсе изучается достаточно кратко и неглубоко, и съемка мультфильма займет много времени, абсолютно ничего нового не дав ни ученику, участвующему в процессе, ни учителю. А вот формулировка задания, выполнение которого свяжет в голове ученика теоретические знания про названия и заряды составляющих атом и ядро частиц и опытные факты:

«Опираясь на знания о строении атома,ними мультфильм, объясняющий, почему при электризации трением заряды, образующиеся на поверхностях, всегда противоположных знаков».

Трудоемкость поиска и формулировки ответа на такой вопрос и трудоемкость получения мультфильма в данном случае соответствуют друг другу, и процесс съемки позволит ответ тщательно обдумать.

Для правильного формирования подходов к оцениванию мультипликационных ответов, как и для разработки успешной постановки вопросов перед учеником, педагог обязан сам уметь создавать мультипликационные образовательные объекты. Однако важно помнить, что самую большую образовательную и всесторонне развивающую пользу приносит не разбор чужого продукта, а работа ученика над своим мультфильмом.

Некоторые **примеры мультфильмов**, сделанных учениками в процессе изучения разных предметов, были опубликованы, и с ними можно познакомиться по ссылке: <http://nmf.multistudia.ru/>

В качестве примеров рассмотрим кадры из мультфильмов учащихся «Работа паровой машины» (физика, VIII класс) и «Питание клетки» (биология, VI класс).

Мультфильм «Работа паровой машины» (рис. 1) — ответ учеников (микрогруппа — три человека) на вопрос, поставленный учителем в момент изучения темы (первичного знакомства). По представленному тексту и рисункам ребятам надо было воспроизвести варианты работы паровой машины.

Мультфильм «Питание клетки» (рис. 2) — ответ учеников (микрогруппа — два человека) на вопрос, поставленный учителем на этапе проверки знаний предыдущего материала. Вопрос был таким: какой вы можете привести пример (в виде мультфильма) осуществления какого-либо процесса жизнедеятельности клетки?

В заключение отметим, что урок с использованием мультипликационных технологий может быть организован учителем по-разному, в зависимости от педагогической обстановки и технической оснащенности. Это могут быть индивидуальные задания, которые в ходе

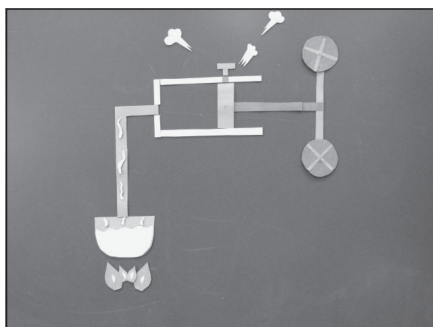
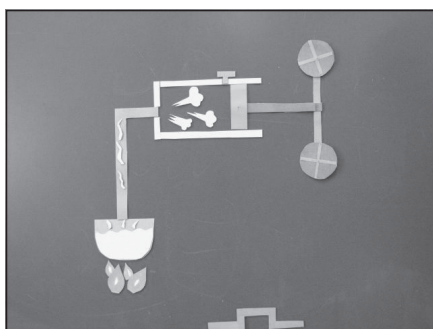
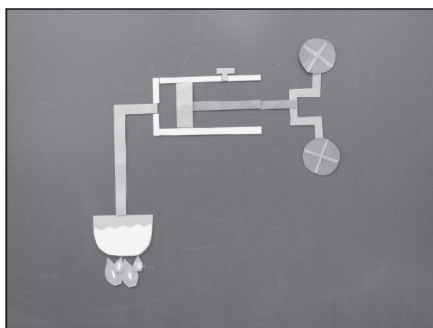
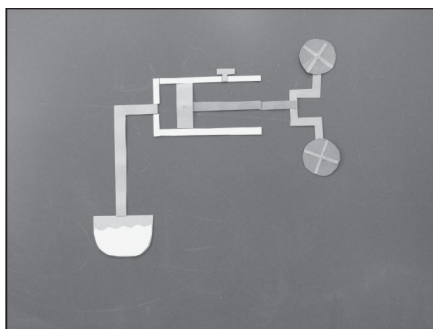


Рис. 1

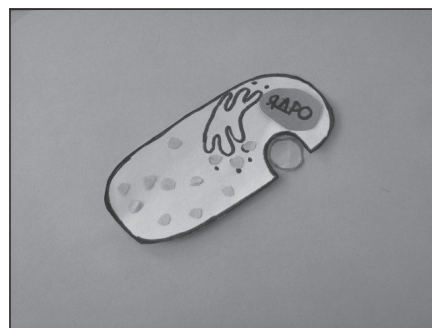


Рис. 2

урока выполняет конкретная группа учеников, или фронтальное применение технологии, когда весь класс делится на группы с индивидуальной задачей для каждой и получение и обсуждение мультипликационных ответов занимает весь урок.

Список использованных источников

1. Арнаутова Н. А. Технология самодельной мультипликации: образовательное пространство школьного предмета и возможности развития когнитивной сферы учащихся // Ребенок в образовательном пространстве мегаполиса: сборник материалов межрегиональной научно-практической конференции (15–16 апреля 2014 года, г. Москва, МГПУ). М.: Экон-информ, 2014.
2. Арнаутова Н. А., Кричевец Е. А. Уроки мультипликации: возможности использования цифровой техники в коррекционно-развивающих целях // Воспитание и обучение детей с нарушениями развития. 2011. № 1.
3. Бабанский Ю. К. Методы обучения в современной общеобразовательной школе. М.: Просвещение, 1985.
4. Лернер И. Я. Дидактические основы методов обучения. М.: Педагогика, 1981.
5. Осипенко Л. Е., Карпова С. И. Педагогические условия использования репродуктивных и исследовательских методов обучения // Вестник Тамбовского университета. Серия «Гуманитарные науки». 2012. № 9.
6. Толлингерова Д., Голоушева Д., Канторкова Г. Психология проектирования умственного развития детей / под ред. В. Я. Ляудис. М.: Роспедагентство, 1994.

В. А. Воробьева,
школа № 962, г. Москва

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ОПЕРЕЖАЮЩЕГО ОБУЧЕНИЯ ПРИ ПРОВЕДЕНИИ ПРАКТИЧЕСКИХ РАБОТ ПО ИНФОРМАТИКЕ В ОСНОВНОЙ ШКОЛЕ

Аннотация

В статье рассматривается технология опережающего обучения («перевернутый класс»), а также особенности ее применения в современном образовательном контексте; описывается методическое обеспечение, разработанное для проведения практических работ по информатике по данной технологии в основной школе.

Ключевые слова: опережающее обучение, информатика, педагогическая технология, обучающие видеоролики, моделирование.

Контактная информация

Воробьева Виктория Александровна, магистрант кафедры теоретической информатики и дискретной математики Московского педагогического государственного университета, учитель информатики школы № 962, г. Москва; *адрес:* 127562, г. Москва, ул. Санникова, д. 3, корп. 3; *телефон:* (499) 201-69-11; *e-mail:* jackybs@mail.ru

V. A. Vorobeva,
School 962, Moscow

APPLICATION OF FLIPPED CLASSROOM STRATEGY FOR REALIZATION OF PRACTICAL TASKS ON INFORMATICS IN SECONDARY SCHOOL

Abstract

The article describes flipped classroom strategy and features of its application in modern educational environment. The article also overviews learning materials, which were developed for conducting practical works as a form of application of flipped classroom strategy in informatics lessons in secondary school.

Keywords: flipped classroom, informatics, pedagogical technology, educational videos, modeling.

Уровень развития технического и программного обеспечения достаточно высок в настоящее время, что позволяет модернизировать уже существующие образовательные методики и технологии таким образом, чтобы образовательный процесс был значительно проще и интереснее для его участников. Одной из таких технологий является **технология опережающего обучения**, предложенная педагогом-новатором С. Н. Лысенковой в конце прошлого века.

Среди характерных особенностей данной технологии в ее первоначальной форме можно выделить [3]:

- личностный подход в педагогике сотрудничества;
- успех — главное условие развития детей в обучении;
- комфортность в классе;
- доброжелательность, взаимопомощь;
- предупреждение ошибок, а не работа над ними;
- последовательность, системность содержания учебного материала;
- дифференциация, доступность задания для каждого;
- к полной самостоятельности приходим постепенно;
- через знающего ученика учить незнающего.

С. Н. Лысенкова обнаружила, что **для повышения эффективности изучения трудных вопросов учебной программы необходимо предварительно и небольшими частями вводить их в учебный процесс**. Тему, вызывающую затруднения, она начинает осваивать с учениками постепенно и гораздо раньше, чем того требует программа [3].

Учащиеся осваивают сложные темы программы в три этапа:

- 1) постепенный ввод основ сложных тем школьной программы предварительно;
- 2) введение, синтез и применение на практике новых понятий;
- 3) развитие беглости мышления в направлении данной темы.

Вторая особенность технологии С. Н. Лысенковой — методический прием, который называется **комментируемое управление** [3]. Выполняя задание с места, ученик проговаривает вслух каждый шаг, что позволяет осуществлять контроль над своей деятельностью и вовлекает в работу всех учеников класса.

Таким образом, одновременно решаются проблемы:

- контроля над собственной деятельностью ученика;
- управления работой всех учащихся данного класса;
- развития учебного диалога.

Третья особенность технологии С. Н. Лысенковой — **обобщающие схемы**, содержащие основы темы и помогающие ученикам при выполнении заданий [3].

Данные схемы в *начальной* школе составляются учителем и используются учениками на уроках и при выполнении домашнего задания до тех пор, пока материал не будет прочно усвоен всеми учениками.

В *основной* школе можно постепенно переходить к составлению таких схем обучающимися в три шага:

- 1) заполнение пустой схемы;
- 2) дополнение схемы;
- 3) самостоятельное построение схемы (ментальной карты).

При использовании данной технологии достигаются все положения парадигмы теоретического мышления. Знания даются системно, последовательно, начиная с основополагающих понятий, компоненты которых рассматриваются с опережением, что обеспечивает их лучшее усвоение.

При применении технологии С. Н. Лысенковой в современном контексте происходит некоторое **изменение ее концептуальных поло-**

жений, связанное с современными техническими возможностями, а также особенностями системно-деятельностного подхода в обучении, которому образовательные тенденции настоящего времени отдают приоритет:

- мотивация через использование социально значимого контекста практических работ;
- в каждой работе интегрируются несколько тем (одни изучены, другие даются с опережением);
- домой задается с опережением работа, которая предварительно обсуждается всеми учениками в классе с учителем;
- при обсуждении работы строится опорная схема с используемыми понятиями, таким образом систематизируется весь используемый для выполнения практической работы материал;
- из данной одинаковой для всех работы может у каждого вырасти свой проект (возможности расширения контекста, исходя из индивидуальных интересов и потребностей обучающегося);
- публичная защита работ для коллективной оценки результата (объективная оценка результата);
- рекомендация к участию в открытом (за пределами школы) конкурсе.

Данная технология нашла свое место в современном образовании как в зарубежных странах, так и в России при преподавании различных предметов [2]. Она является одной из наиболее эффективных современных педагогических технологий.

Мы решили **применить данную технологию при проведении компетентностно-ориентированных практических работ по информатике в основной школе**. Апробация проходила на традиционных темах для информатики: использование электронных таблиц для решения задач табулирования функции, составления штатного расписания организации, визуализации данных, сортировки и фильтрации данных. Потенциал работ такого типа, когда решается практико-ориентированная задача и в ходе ее решения изучаются возможности программного продукта, достаточно высокий, но ощущается недостаток доступных учителю методических разработок.

Нами было разработано **методическое обеспечение для проведения практических работ** по информатике в основной школе для освоения возможностей электронных таблиц как эффективного средства моделирования. В него входит практикум, содержащий семь тематических работ. Каждая работа состоит из:

- кратких обучающих видеороликов, которые содержат теоретический материал для самостоятельного изучения в качестве домашнего задания;
- практической работы для проведения на уроке;
- контрольного листа для проверки результатов обучения.

Обучающие видеоролики длятся не более семи минут и содержат примеры ситуаций, в которых могут потребоваться знания и способы деятельности, получаемые в ходе изучения данной темы (рис. 1).

Также видеоролики рассматривают сам ход решения данной задачи.

Работы, предназначенные для проведения на уроке, опираются на материалы видеоролика с объяснением решения, которое было рассмотрено дома самостоятельно.

Организация работы происходит следующим образом:

- 1) В качестве домашнего задания обучающиеся получают теоретический материал в форме видеоро-



Рис. 1

ликов, подкрепленный простыми практическими заданиями, ход решения которых также описан в видеороликах (рис. 2). Например, задача на табулирование функции может быть задачей по физике на нахождение скорости падения тела.

- 2) На уроке происходит обсуждение изученного самостоятельно материала, а также составление опорных схем по каждой теме.
- 3) После обсуждения обучающиеся приступают к выполнению практических работ.

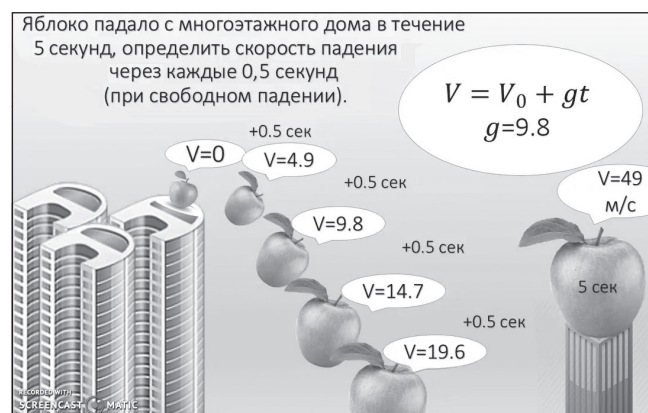


Рис. 2

Приведем **пример постановки задачи** для изучения видов адресации в Excel:

«Робот находится в точке с координатами (1; 2). Каждую минуту он перемещается на 0,2 см вправо и на 0,5 см вверх и останавливается. Робот начал свое движение в 13:30. Определить произведение его координат для каждой точки его остановки на интервале [(1; 2) — (3; 7)]. Сколько остановок сделает робот в период времени с 13:35 до 13:40 включительно?»

Ход работы над задачей по технологии опережающего обучения.

Учащийся получает в виде домашнего задания видеоролик с объяснением решения задачи, которое включает необходимый теоретический материал (рис. 3). В видеоролике пошагово объясняется и выполняется задание.

При обсуждении этого задания в классе составляется опорная схема по теме в виде ориентированного графа.

После коллективного обсуждения постановки задачи обучающиеся выполняют практическую работу (рис. 4)

Яблоко падало с многоэтажного дома в течение 5 секунд, определить скорость падения через каждые 0,5 секунд (при свободном падении)

$V = V_0 + gt$
 $g = 9.8$

Относительная - это адресация, при которой формула читается относительно текущего положения ячейки.

Адресация - это указание на данные находящиеся в определённой ячейке.

Абсолютная - это адресация, при которой идёт указание на конкретную ячейку, адрес которой не изменяется. \$ - признак абсолютной адресации

Смешанная - адресация, при которой идёт изменение только одного адреса. Знак \$ ставится только в одном месте, или перед буквой столбца или перед номером строки.

Рис. 3

	A	B	C	D	E	F	G	H	I	J
1	Основная таблица					Вспомогательная таблица				
2	X	Y	F=X*Y	T		X0	step X	Y0	step Y	
3	1	2	2	13:30		1	0,2	2	0,5	
4	1,2	2,5	3	13:31						
5	1,4	3	4,2	13:32						
6	1,6	3,5	5,6	13:33						
7	1,8	4	7,2	13:34						
8	2	4,5	9	13:35						
9	2,2	5	11	13:36						
10	2,4	5,5	13,2	13:37						
11	2,6	6	15,6	13:38						
12	2,8	6,5	18,2	13:39						
13	3	7	21	13:40						
14			Сумма значений F	110						
15			Количество остановок с 13:35 до 13:40	6						

Рис. 4

и сдают результат выполнения учителю по контрольному листу.

Контрольный лист содержит основные понятия и умения по изученной теме в виде опросника [5]:

«Проверьте, знаете ли вы, что такое:

- табулирование функций;
- форматы чисел;
- умеете ли вы:
- применять функции;
- менять форматы представления чисел;
- защищать информацию рабочего листа и рабочей книги».

Учитель опрашивает учеников по контрольным листам и оценивает результаты работы.

Такая технология позволяет учащимся быть всегда включенными в образовательный процесс и завершать выполнение практических работ вовремя и с положительной оценкой. Таким образом на уроке освобождается

время для решения более сложных заданий под контролем учителя, а теоретический материал рассматривается самостоятельно и представлен в форме, интересной для обучающихся.

Список использованных источников

1. Воронцов С. Г. Общеучебные умения как деятельностный компонент содержания учебно-познавательной компетенции // Центр дистанционного образования «Эйдос». <http://www.eidos.ru/journal/2007/0930-9.htm>
2. Лапчик М. П., Семакин И. Г., Хеннер Е. К., Самылкина Н. Н. Теория и методика обучения информатике. М.: Академия, 2008.
3. Лысенкова С. Н. Методом опережающего обучения. М.: Просвещение, 1988.
4. Программы для общеобразовательных учреждений: Информатика. 2–11 классы / сост. М. Н. Бородин. 2-е изд., испр. и доп. М.: БИНОМ. Лаборатория знаний, 2005.
5. Самылкина Н. Н. Современные средства оценивания результатов обучения. М.: БИНОМ. Лаборатория знаний, 2007.

В. В. Тарапата,
школа № 283, г. Москва

ЛАБОРАТОРНАЯ РАБОТА ПО ОСНОВАМ МИКРОЭЛЕКТРОНИКИ «АСИНХРОННЫЙ RS-ТРИГГЕР» В УГЛУБЛЕННОМ КУРСЕ ИНФОРМАТИКИ ДЕСЯТОГО КЛАССА*

Аннотация

В статье представлена лабораторная работа по созданию модели асинхронного RS-триггера для учащихся десятого класса, изучающих информатику на углубленном уровне. Необходимость такой работы обусловлена чисто теоретическим характером изучения тем «Логические элементы компьютера» и «Микроэлектронные устройства», отсутствием наглядных пособий.

Ключевые слова: информатика, основы микроэлектроники, логические элементы компьютера, триггер, прототипирование, наглядное пособие.

Контактная информация

Тарапата Виктор Викторович, аспирант Московского педагогического государственного университета, учитель информатики школы № 283, г. Москва; адрес: 127224, г. Москва, ул. Широкая, д. 21А; телефон: (499) 477-11-40; e-mail: vvteach@mail.ru

V. V. Tarapata,
School 283, Moscow

LABORATORY RESEARCH IN BASIC COURSE OF MICROELECTRONICS "ASYNCHRONOUS RS-TRIGGER" IN ADVANCED INFORMATICS COURSE IN THE 10TH GRADE

Abstract

The article describes a laboratory research that is devoted to the creation of a pattern of asynchronous RS-trigger for pupils in the tenth grade who study advanced course of informatics. The relevance and necessity of this work is conditioned by purely theoretical character of studies on the topics "Logical computer elements" and "Microelectronic devices" as well as by the absence of teaching aids.

Keywords: informatics, basic course of microelectronics, computer logic elements, trigger, prototyping, visual aids.

Как правило, изучение тем «Логические элементы компьютера» и «Микроэлектронные устройства» даже на углубленном уровне изучения информатики в старшей школе носит сугубо теоретический характер [1, 3, 4] и характеризуется отсутствием наглядных пособий.

Основной целью работы «Асинхронный RS-триггер» является практическое знакомство учащихся с основами микроэлектроники в процессе сборки асинхронного RS-триггера, построенного на универсальных логических элементах ИЛИ—НЕ.

Триггер — это электронная схема, имеющая два устойчивых состояния, одно из которых принимается за логический ноль «0», а другое — за логическую единицу «1». Под действием управляющих сигналов на входе триггер скачкообразно переходит из одного состояния в другое («опрокидывается»). Таким образом, один триггер способен хранить 1 бит информации.

Триггер служит основой создания регистров, счетчиков и различных запоминающих устройств в вычислительных устройствах.

В данной лабораторной работе исследуется принцип работы **асинхронного RS-триггера** — компонента, который может находиться в одном из двух устойчивых состояний и управляется двумя входными сигналами:

R (от *англ.* Reset — сброс) — переключение триггера в состояние «0»;

S (от *англ.* Set — установка) — переключение триггера в состояние «1».

Составим таблицу истинности, описывающую логику работы асинхронного RS-триггера:

R	S	Q_{n-1}	$\rightarrow$	Q_n	Режим
0	0	0		0	Хранение информации
0	0	1		1	
0	1	0		1	Запись «1»
0	1	1		1	
1	0	0		0	Запись «0»
1	0	1		0	
1	1	0		X	Запрещенное состояние (неопределенность)
1	1	1		X	

Здесь:

R, S — входные сигналы;

Q_{n-1} — предыдущее состояние триггера (до подачи сигналов);

Q_n — последующее состояние триггера (после подачи сигналов).

Асинхронный RS-триггер возможно создать на базе элементов ИЛИ—НЕ. Изобразим принципиальную электрическую схему (рис. 1).

Заметим, что для работы триггера необходим не только активный выход, который совпадает с состоянием триггера (Q), но и инверсный выход ($\bar{Q}$). Таким образом, триггер может «понимать» свое предыдущее состояние. Для лучшего понимания можно самостоятельно рассмотреть его реализацию на биполярных транзисторах.

Построенное устройство называется «асинхронный RS-триггер» и имеет следующее условное графическое обозначение (рис. 2).

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_08-2017/

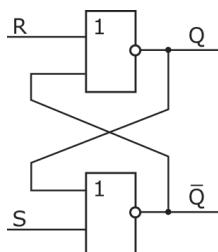


Рис. 1. Принципиальная электрическая схема

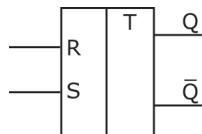


Рис. 2. Условное графическое обозначение RS-триггера

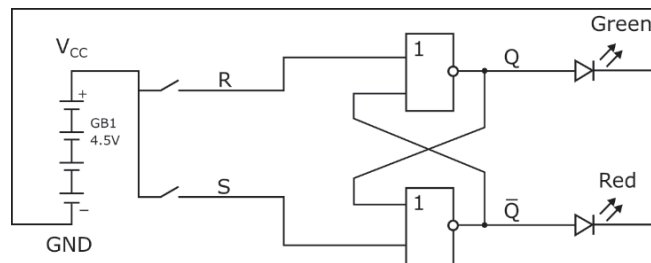


Рис. 3. Принципиальная электрическая схема модели RS-триггера

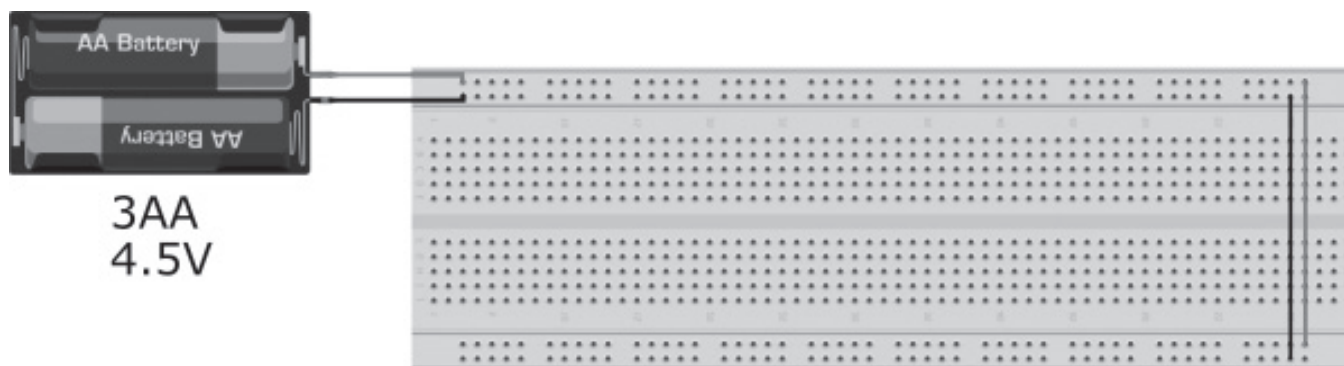


Рис. 4

Отметим, что триггер, построенный на элементах ИЛИ—НЕ, работает на уровне высоких сигналов, т. е. его активный уровень — это логическая единица «1». Такой триггер удобен для использования в демонстрационных целях. На практике возможна реализация триггера на элементах И—НЕ, но его активный уровень будет совпадать с логическим нулем «0», т. е. он работает на уровне низких сигналов, что неудобно для демонстрации.

Кстати! У триггера существует еще одно название — «бистабильная ячейка» (компонент, который может находиться в двух стабильных состояниях). Различают дизъюнктивную бистабильную ячейку (на элементах ИЛИ—НЕ) и конъюнктивную бистабильную ячейку (на элементах И—НЕ).

На основе логической схемы запишем **характеристическое уравнение асинхронного RS-триггера**, построенного на элементах ИЛИ—НЕ:

$$Q_n = R + \overline{Q_n} = R + \overline{Q_{n-1}} + S = R + \overline{Q_{n-1}} \cdot S = \overline{R} \cdot (Q_{n-1} + S).$$

Наша **задача**: создать электрическое устройство, иллюстрирующее принцип работы асинхронного RS-триггера.

Рассмотрим **логику его работы**:

1. Подача входных сигналов R и S будет осуществляться нажатием соответствующей кнопки, причем если кнопка нажата, то подается сигнал «1», если не нажата — сигнал «0».
2. Выходные сигналы будут проиллюстрированы с помощью двух светодиодов — зеленого для активного выхода Q и красного для инверсного выхода.

На основании данной логики изобразим **принципиальную схему устройства** (рис. 3).

Оборудование для лабораторной работы:

- макетная плата Breadboard 830 — 1 шт.;
- батарейный отсек 3×AA — 1 шт.;
- элемент питания AA — 3 шт.;

- микросхема стандартной логики: SN74HC02N (4×2 ИЛИ—НЕ) — 1 шт.;
- кнопка с фиксацией — 2 шт.;
- светодиоды — 2 шт.;
- комплект проводов-перемычек «папа—папа» — 1 уп.

Ход лабораторной работы.

1. Перед началом работы убедитесь, что переключатель на батарейном отсеке стоит в положении OFF. Вставьте три элемента питания AA в батарейный отсек.

2. Сначала обеспечьте макетную плату питанием и «землей» (рис. 4).

- Подключите **красный** провод батарейного отсека к шине, обозначенной красной линией.
- Для того чтобы питание было доступно с двух сторон макетной платы, соедините верхнюю шину питания с нижней красным проводом.
- Аналогично подключите черный провод («землю») на шины, обозначенные синей линией.

Внимание! В случае, если макетная плата имеет «разрывы» в других местах, точно так же перекиньте питание и «землю» с помощью проводов.

3. Смонтируйте на макетной плате две кнопки, как показано на рисунке 5.



Рис. 5

Таким образом, каждый из контактов кнопок будет независим. Сверху и снизу обязательно должны остаться свободные пины*.

4. К каждой кнопке подведите питание и заземлите кнопки (рис. 6).

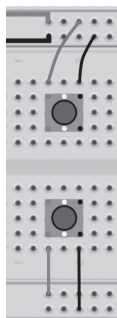


Рис. 6

5. Посередине платы установите два светодиода — зеленый и красный — и заземлите их (рис. 7).



Рис. 7

Зеленый светодиод говорит нам о состоянии триггера: если он горит, то триггер хранит «1», если нет — триггер хранит «0». Соответственно, красный светодиод всегда хранит противоположное значение.

6. Смонтируйте микросхему SN74HC02N (ИЛИ–НЕ), как показано на рисунке 8.

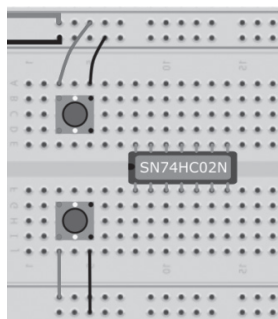


Рис. 8

7. Согласно DataSheet подключите питание (пин № 14) и «землю» (пин № 7) (рис. 9).

8. Расположенные слева кнопки будут отвечать за подачу сигналов: верхняя — за сигнал R (Reset), нижняя — за сигнал S (Set).



Рис. 9

9. Тот вход микросхемы, который напрямую подключается к R , будем называть *обработчиком сигнала R* . Вход, который напрямую подключается к S , будем называть *обработчиком сигнала S* .

10. Пора приступить к подключению всех компонентов согласно принципиальной схеме. Старайтесь использовать одноцветные провода-перемычки для одних и тех же каналов.

Выход R подключите к входу 11 «ИЛИ–НЕ» (рис. 10).

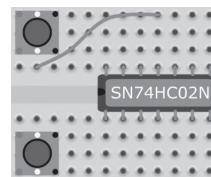


Рис. 10

Сигнал из выхода 13 «ИЛИ–НЕ» будет в дальнейшем подаваться на зеленый светодиод, но светодиоды мы подключим в самом конце.

11. Выход S подключите к входу 2 «ИЛИ–НЕ» (рис. 11).

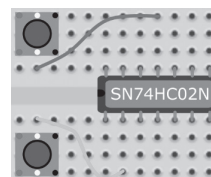


Рис. 11

12. Результат обработки сигнала S необходимо отправить на вход обработчика сигнала R . Для этого соедините выход 1 «ИЛИ–НЕ» с входом 12 «ИЛИ–НЕ» (рис. 12).

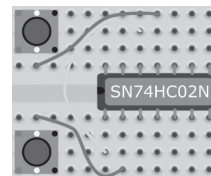


Рис. 12

13. Аналогично результат обработки сигнала R необходимо отправить на вход обработчика сигнала S . Для этого соедините выход 13 «ИЛИ–НЕ» с входом 3 «ИЛИ–НЕ» (рис. 13).

* Пин (от *англ.* pin — штырек) — отверстие на макетной плате.

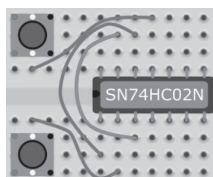


Рис. 13

14. Остается подключить светодиоды. Зеленый светодиод нужно соединить с выходом обработчика сигнала R . Подключите выход 13 «ИЛИ–НЕ» к зеленому светодиоду (рис. 14).

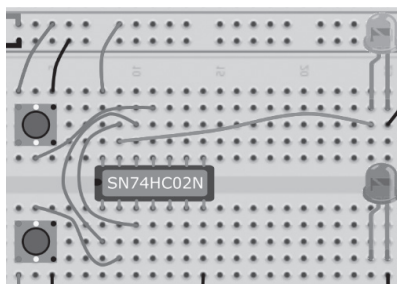


Рис. 14

15. Аналогично подключите красный светодиод. Соедините выход 1 «ИЛИ–НЕ» с красным светодиодом (рис. 15).

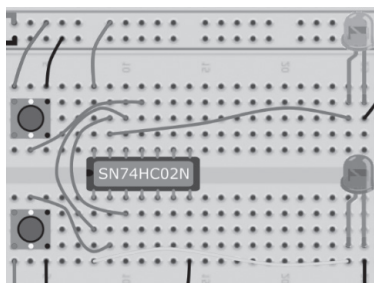


Рис. 15

Проверьте еще раз все подключения и расположение элементов согласно общей схеме.

Приложения

Схема соединения шин макетной платы



Рис. 17

Задания.

1. Проведите эксперименты, проверив все состояния триггера, и заполните таблицу истинности.

Исходное состояние		R	S	Новое состояние		Режим
Q (Green)	$\bar{Q}$ (Red)			Q (Green)	$\bar{Q}$ (Red)	
1	0	0	0			
0	1	0	0			
1	0	0	1			
0	1	0	1			
1	0	1	0			
0	1	1	0			
1	0	1	1			
0	1	1	1			

2. Рассмотрите принципиальную схему асинхронного RS-триггера, построенного на элементах «И–НЕ» (рис. 16), и выведите для него характеристическое уравнение.

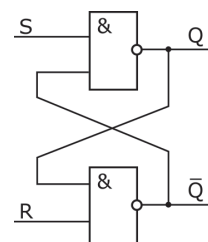


Рис. 16

$$Q_n =$$

3. Каким будет принцип действия асинхронного RS-триггера, построенного на элементах «И–НЕ», относительно построенного на элементах «ИЛИ–НЕ»: обратным, противоположным, контрапозитивным?

Ответ:

Приложение 1

Общий вид модели асинхронного RS-триггера

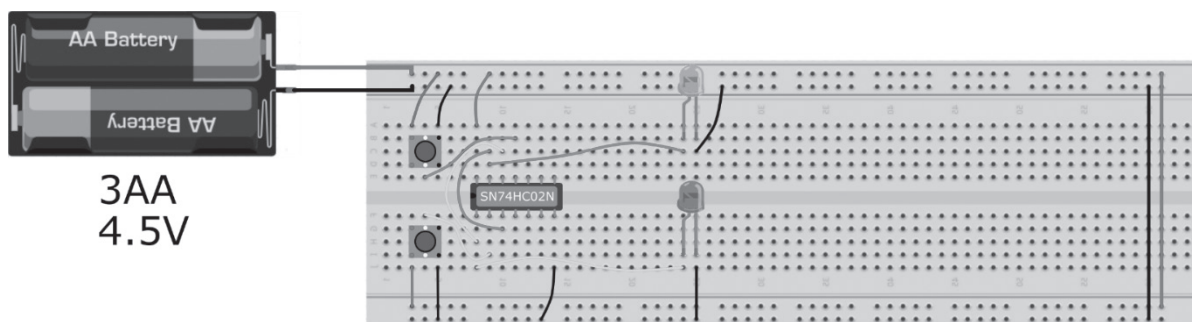


Рис. 18

Приложение 3

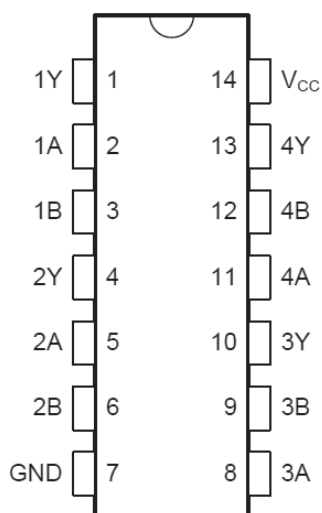
DataSheet Texas Instruments SN74HC02N
«ИЛИ–НЕ»

Рис. 19

Контакты элемента «Кнопка с фиксацией»

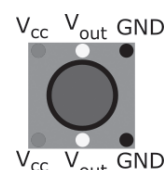


Рис. 20

Приложение 5

Контакты элемента «Светодиод»



Рис. 21

Список использованных источников

1. Андреева Е. В., Босова Л. Л., Фалина И. Н. Математические основы информатики. М.: БИНОМ. Лаборатория знаний, 2007.
2. Королева Л. В., Петрова Е. Б. Основы микроэлектроники: учебное пособие для студентов педагогических вузов. М.: Карпов Е. В., 2016.

3. Поляков К. Ю., Еремин Е. А. Информатика: учебник для 10 класса: в 2 ч. Ч. 1 (углубленный уровень). М.: БИНОМ. Лаборатория знаний, 2016.
4. Самылкина Н. Н., Калинин И. А. Информатика: учебник для 10 класса (углубленный уровень). М.: БИНОМ. Лаборатория знаний, 2014.
5. «Чип и Дип». <https://www.chipdip.ru/>
6. Texas Instruments. <http://www.ti.com/>

М. Ю. Глотова, Е. А. Самохвалова,
Московский педагогический государственный университет

ИСПОЛЬЗОВАНИЕ POWERPOINT ДЛЯ ВИЗУАЛИЗАЦИИ ИНФОРМАЦИИ

Аннотация

В статье рассматриваются вопросы обучения на уроках информатики визуализации информации средствами презентационной графики на примере MS PowerPoint. Обучение навыкам визуализации развивает умения использовать функционально-графические представления информации в процессе учебы в образовательном учреждении и служит для прикладного использования обучающимися в их дальнейшей деятельности.

Ключевые слова: информатика, MS PowerPoint, презентационная графика, визуализация информации.

Контактная информация

Глотова Марина Юрьевна, канд. физ.-мат. наук, доцент, зав. кафедрой информационных технологий в образовании Московского педагогического государственного университета; *адрес:* 119571, г. Москва, пр-т Вернадского, д. 88, каб. 732; *телефон:* (495) 438-11-53; *e-mail:* myu.glotova@mpgu.edu

Самохвалова Евгения Александровна, ст. преподаватель кафедры информационных технологий в образовании Московского педагогического государственного университета; *адрес:* 119571, г. Москва, пр-т Вернадского, д. 88, каб. 732; *телефон:* (495) 438-11-53; *e-mail:* ea.samokhvalova@mpgu.edu

M. Yu. Glotova, E. A. Samokhvalova,
Moscow State University of Education

USING POWERPOINT FOR VISUALIZATION OF INFORMATION

Abstract

The article deals with the issues of teaching visualization of information at the lessons of informatics with means of presentation graphics on the example of MS PowerPoint. Training in visualization skills develops the ability to use functional-graphical representations of information in the process of studying in an educational institution and serves for applied use by students in their future activity.

Keywords: informatics, MS PowerPoint, presentation graphics, visualization of information.

Под *визуализацией* понимается любой способ обеспечения наблюдаемости реальности, а под *результатом визуализации*, или *визуальной моделью*, — любая зрительно воспринимаемая конструкция, имитирующая сущность объекта познания. Особенно эффективно методы визуализации используются для представления в наглядном виде изначально незрительной информации (например, такой как учет успеваемости учащихся, распределение плотности населения, грамматические структуры определенного языка и т. д.). Грамотно выполненная визуализация помогает быстро и эффективно донести ключевое сообщение до аудитории.

Визуализацию информации в современном образовательном процессе можно рассматривать как реализацию одного из важнейших дидактических принципов — принципа наглядности в обучении. В то же время здесь затрагивается и принцип природосообразности обучения, поскольку визуализация информации отвечает потребностям большей части обучаемых в преобладающем восприятии информации через зрительный анализатор.

С одной стороны, обучение навыкам визуализации призвано развивать у учащихся умение использовать функционально-графические представления информации в процессе учебы в образовательном учреждении, а с другой стороны, эти навыки пригодятся обучающимся для прикладного использования в их дальнейшей деятельности.

В данной статье мы рассмотрим **примеры практических заданий в рамках изучения работы с презентационной графикой в программе MS PowerPoint 2010 и выше**. Представленный набор заданий рассчитан на два урока с выполнением части заданий в качестве домашней работы.

Урок 1

Задание 1. Создание диаграмм и графиков.

Для того чтобы в MS PowerPoint добавить на слайд диаграмму, необходимо выполнить следующие действия.

1. Перейдите на вкладку *Вставка*.

2. В группе *Иллюстрации* щелкните на кнопке *Диаграмма*. Откроется диалоговое окно *Вставка диаграммы*. В правой части данного окна перечислены виды диаграмм, а в левой — представлены миниатюры различных вариантов диаграмм.

По умолчанию предлагают стандартную столбчатую диаграмму, пока пусть она и остается.

3. Выберите подходящий вариант диаграммы и щелкните на кнопке *ОК*.

Вид окна и панели изменился, теперь на экране видны рабочие поля сразу двух программ — PowerPoint и Excel. Для примера нам предлагают таблицу, в которой уже есть значения. Их 12 — три столбца и четыре строки. На основании этих табличных данных и построена диаграмма (рис. 1).

Строк и столбцов в таблице может быть любое количество: можно ниже и правее добавить еще данные, а названия строк и столбцов изменить так, как нам необходимо.

Границей синего цвета на листе Excel обозначена область данных для построения графика или диаграммы. Если в эту границу не попадают все необходимые нам данные, растягиваем обозначенную область так, как нам нужно, воспользовавшись синим маркером, который виден в правом нижнем углу выделенной области. Если подвести к нему курсор, вместо крестообразного он примет вид двунаправленной стрелки. Нажимаем левую кнопку мыши и растягиваем область так, чтобы все нужные данные оказались внутри обозначенной границы (рис. 2).

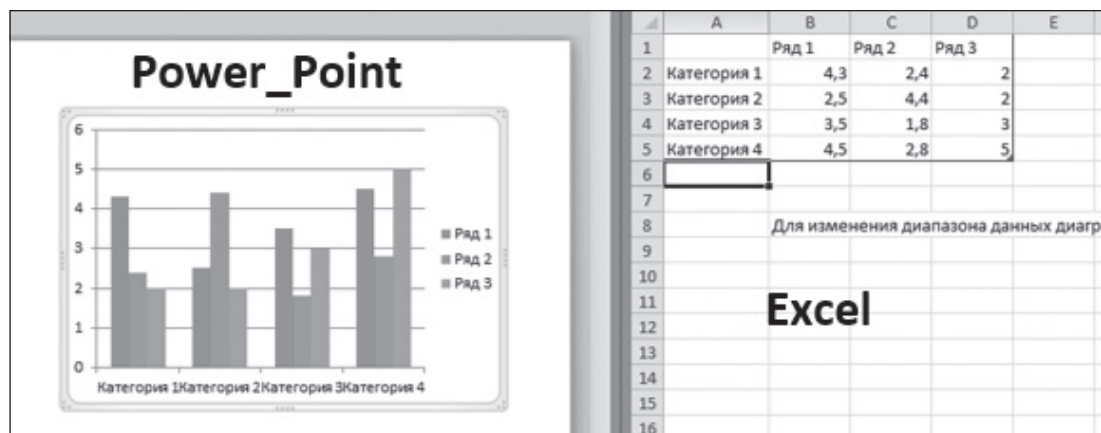


Рис. 1

	А	В	С	Д	Е	Ф
1		Математика	Рус.яз	Ин.яз	Биология	Физ-ра
2	Васильев	4	3	4	5	5
3	Трегубов	4	3	4	5	4
4	Потапова	3	4	5	4	4
5	Жмаева	5	5	4	4	5
6	Силаев	4	4	5	5	4
7	Круглова	4	3	4	5	3
8						

Рис. 2

Одновременно на слайде мы получаем вот такой результат (рис. 3):

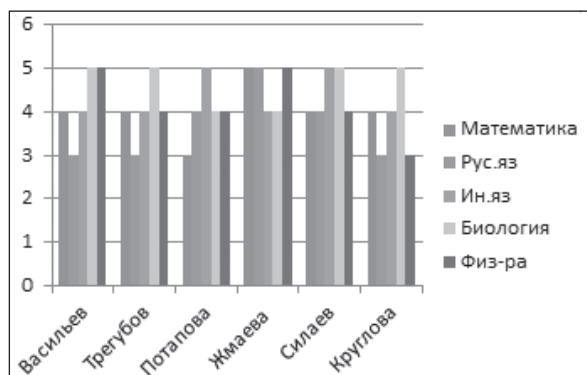


Рис. 3

Теперь перейдем к редактированию внешнего вида диаграммы.

К работе с диаграммами и графиками относятся три вкладки: *Конструктор*, *Макет* и *Формат*. Если диаграмма выделена, эти вкладки отображаются на ленте крайними справа.

4. Закройте программу Excel.

5. Перейдите на вкладку *Конструктор*.

В ней четыре области: *Тип*, *Данные*, *Макет* и *Стили*.

6. *Стили* — это визуальное оформление нашей диаграммы.

Выберите наиболее подходящий вариант оформления вашей диаграммы.

7. Со стилем определились, перейдем к области *Макеты диаграмм*, это следующая область в панели *Конструктор* (рис. 4).



Рис. 4

Здесь также есть возможность выбора. Щелчок на кнопке выбора других вариантов покажет все доступные макеты. Активный, т. е. используемый в текущий момент, макет не подсвечен, он просто отображается первым. Изменяя макет, мы изменяем взаимное расположение всех объектов диаграммы в области построения.

Можно расположить легенду (пояснение к отображаемым данным) сверху, снизу, слева от самой области построения (рис. 5):

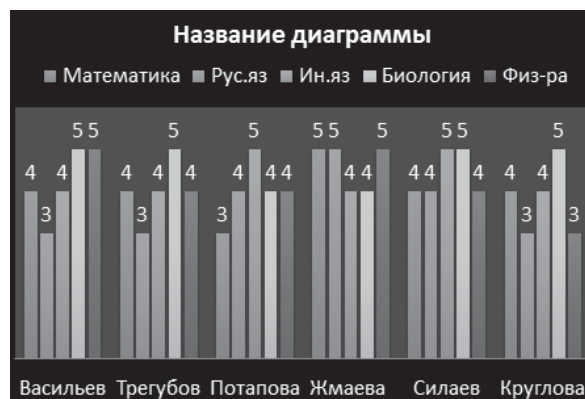


Рис. 5

Выберите *Макет 3*.

8. С помощью кнопки *Изменить тип диаграммы* можно изменить тип диаграммы. Измените тип диаграммы на *Линейчатая*.

9. Перейдите на вкладку *Макет*. Здесь можно отредактировать все компоненты.

10. Первая область на панели *Макет* — *Текущий фрагмент*.

В выпадающем меню *Область диаграммы* области *Текущий фрагмент* выберите для изменения фрагмент *Легенда*. Он будет выделен на слайде.

11. Щелкните на кнопке *Формат выделенного фрагмента* области *Текущий фрагмент*.

Откроется окно изменения параметров того фрагмента, с которым вы работаете в текущий момент, в нашем случае это легенда (набор параметров в окне *Фрагмент...* для разных элементов разный).

Измените расположение легенды и установите градиентную заливку.

12. В выпадающем меню *Область диаграммы* области *Текущий фрагмент* выберите *Область диаграммы* и измените цвет заливки: текстуру, тип линий и тень.

13. В выпадающем меню *Область диаграммы* области *Текущий фрагмент* выберите теперь любой ряд и измените отображения столбцов: задайте границы и зазор между столбцами.

14. Область *Вставка* панели *Макет* позволяет добавить в область диаграммы какой-то рисунок, автофигуру.

15. Область *Подписи* панели *Макет* позволяет оформить надписи в диаграмме.

Так, кнопка *Название диаграммы* откроет меню с выбором варианта отображения заголовка.

Добавьте заголовок в область диаграммы и разместите его над диаграммой.

16. Измените расположение названий вертикальной и горизонтальной осей (рис. 6).



Рис. 6

17. Перейдите на вкладку *Формат*. Здесь можно дополнительно изменить внешний вид диаграммы. Вкладка выглядит почти так же, как при работе с изображениями.

Область *Стили фигур* вкладки *Формат* предназначена, например, для выбора цвета столбцов.

Область *Стили WordArt* вкладки *Формат* предназначена для изменения надписей.

18. После внесения всех изменений можно вернуться на вкладку *Конструктор* и снова изменить тип диаграммы.

Измените внешний вид и тип диаграммы согласно образцу (рис. 7).

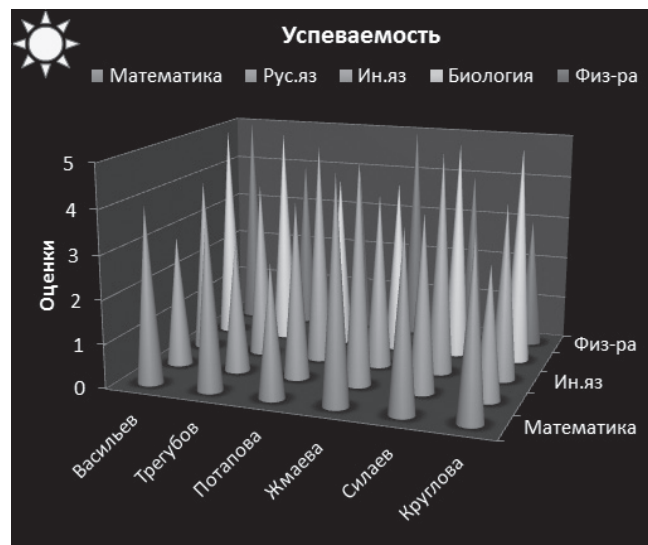


Рис. 7

Задание 2. Анимация диаграммы.

1. Выделите диаграмму.

2. Перейдите на вкладку *Анимация*, выберите в разделе в разделе анимации категорию *Эффекты входа* — *Случайные полосы*.

3. В разделе *Параметры эффектов* выберите: *Направление* — *По вертикали*, *Последовательность* — *По элементам категорий* (рис. 8).

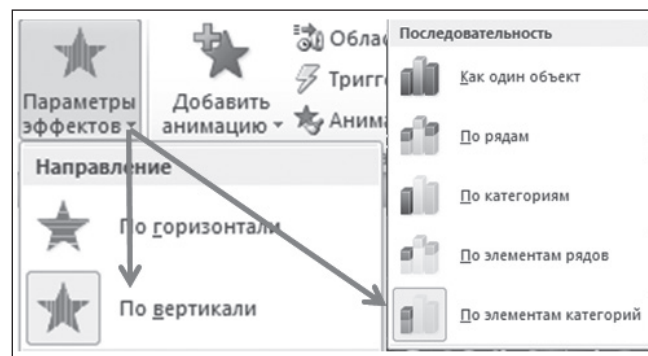


Рис. 8

4. На вкладке *Анимация* нажмите кнопку *Область анимации*.

5. Правой кнопкой мыши щелкните в списке настроек анимации и в контекстном меню выберите пункт *Параметры эффектов*.

6. На вкладке *Эффект* можно задать направление анимации и звуковые эффекты.

Задайте любой звуковой эффект.

7. На вкладке *Время* можно задать параметры хронометража, включая начало анимации, скорость ее воспроизведения, а также время задержки. Задайте: *Начало* — *После предыдущего*, *Продолжительность* — *Очень медленно*, *Повторение* — 2.

8. На вкладке *Анимация диаграммы* можно указать способ анимации каждого элемента диаграммы. Для того чтобы в начале анимации отображалась пустая диаграмма, снимите флажок *Запустить анимацию, нарисовав фон диаграммы* (рис. 9).

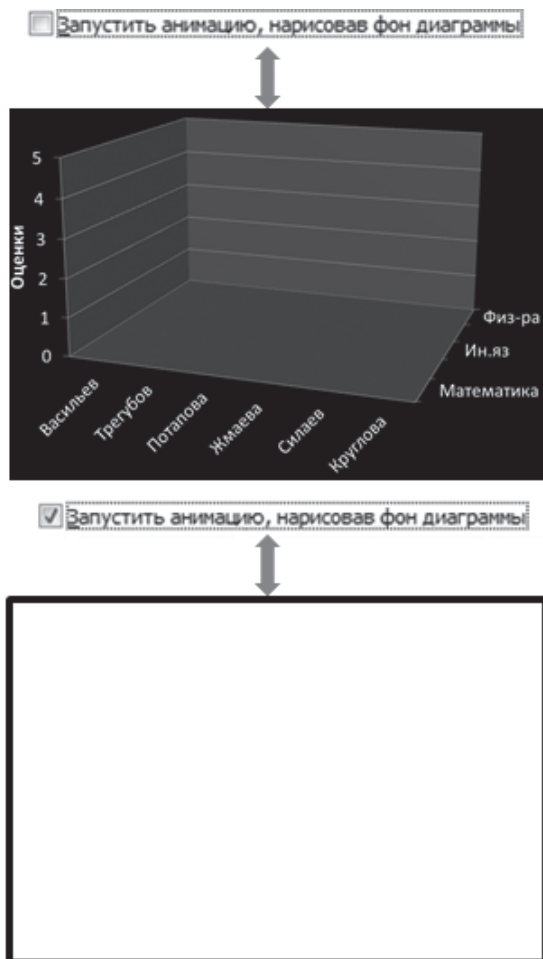


Рис. 9

9. Запустите демонстрацию презентации и оцените результат проделанной работы: вкладка *Показ слайдов*, кнопка *С текущего слайда*.

Домашнее задание.

Самостоятельно создайте три слайда и на каждом из них разместите следующие диаграммы:

Слайд 1. Имеются данные об увлечениях группы людей (рис. 10):

	A	B
1		Проценты
2	Волейбол	26%
3	Математика	17%
4	Теннис	21%
5	Музыка	26%

Рис. 10

Постройте на их основе круговую диаграмму (рис. 11):



Рис. 11

Слайд 2. Имеются данные, характеризующие ввод в действие образовательных учреждений (рис. 12):

	A	B	C	D
1	Столбец1	2012 г.	2013 г.	2014 г.
2	Дошкольные	7	5	13
3	Общеобраз.	54	26	37
4	Школы-интернаты	20	18	9
5	Профессионал.	2	3	5

Рис. 12

Постройте на их основе линейный график (рис. 13):



Рис. 13

Слайд 3. Имеются данные распределения групп крови в трех странах (рис. 14):

	A	B	C	D	E
1	Группа крови	O(I)	A(II)	B(III)	AB(IV)
2	Россия	33	37	23	7
3	Австралия	32	25	18	25
4	Франция	24	19	28	29

Рис. 14

Постройте на их основе диаграмму с областями (рис. 15):

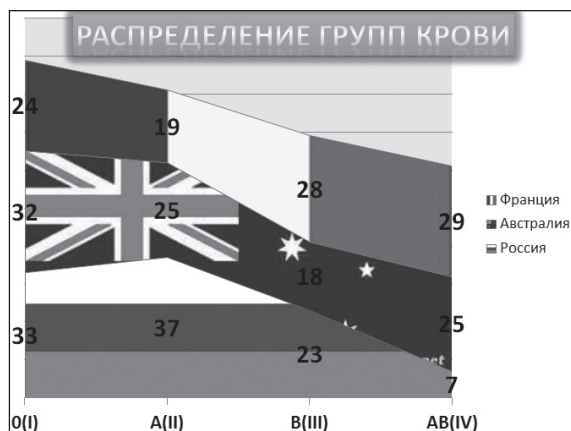


Рис. 15

Задайте различные эффекты анимации для каждой созданной диаграммы, установите дополнительные параметры.

Урок 2

SmartArt — это графические объекты, схемы, которые служат для структурирования текста и изображений для эффектной и наглядной подачи информации. Главное — правильно выбрать объект SmartArt, чтобы он соответствовал содержанию и теме презентации.

Задание 1.

Для того чтобы создать объект SmartArt, выполните следующие действия.

1. Перейдите на вкладку *Вставка*.
2. В группе *Иллюстрации* щелкните на кнопке *SmartArt*. Откроется диалоговое окно *Выбор рисунка SmartArt*.
3. В левой части диалогового окна *Выбор рисунка SmartArt* выберите тип рисунка *Иерархия*. В средней части окна *Выбор рисунка SmartArt* появятся несколько вариантов схем иерархической структуры.
4. Щелкните на первом варианте. Обратите внимание на информацию в правой части окна — там приведены рекомендации о применении выбранного объекта SmartArt.
5. Щелкните на кнопке *ОК*. На слайде появится схема.
6. Заполните схему текстом. Для этого щелкните на стрелке слева от схемы и введите необходимые данные (рис. 16).

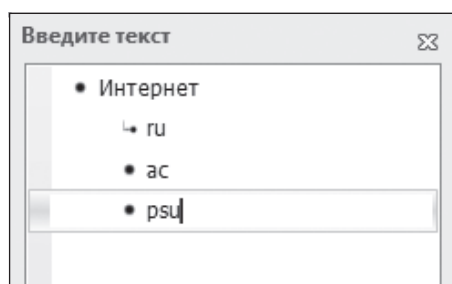


Рис. 16

7. Формат текстовых фрагментов можно изменить. Для этого в схеме на слайде выделите нужный фрагмент текста, задайте новые шрифт и размер, добавьте полужирное начертание.

8. Измените цвет схемы: вкладка *Конструктор*, раздел *Изменить цвета*.

9. Измените вид схемы: вкладка *Конструктор*, раздел *Макеты*.

10. Измените стиль отображения схемы: вкладка *Конструктор*, раздел *Стили SmartArt*.

11. Произведите изменения макета, цвета и стиля для получения схемы, изображенной на рисунке 17.

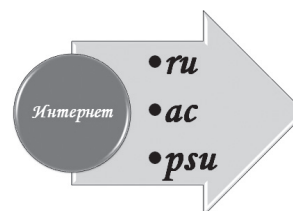


Рис. 17

Задание 2.

1. Создайте еще один слайд.
2. Перейдите на вкладку *Вставка*.
3. В области *Иллюстрации* щелкните на кнопке *SmartArt*.
4. В левой части диалогового окна *Выбор рисунка SmartArt* выберите тип рисунка *Office.com*.
5. Щелкните на варианте *Организационная с рисунком* и заполните схему (рис. 18).
6. Для вставки изображений щелкните левой кнопкой мыши на ячейке В открывшемся окне *Вставка рисунка* выберите изображение и щелкните на кнопке *Вставить* или выделите фрагмент, перейдите на вкладку *Формат*, откройте меню *Заливка фигуры* и выберите *Рисунок* (рис. 18).

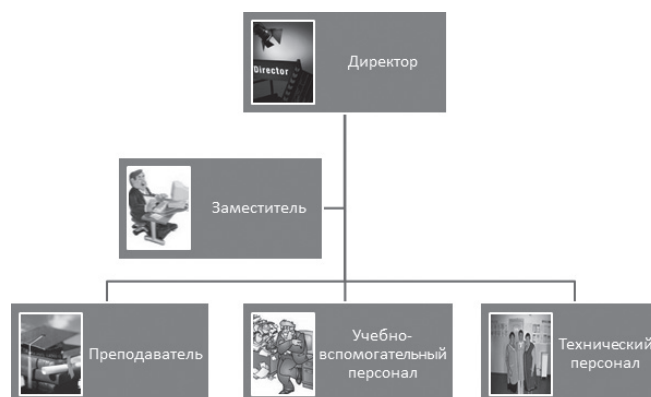


Рис. 18

7. Добавьте на схему еще одну фигуру. Для этого выделите фигуру, после которой необходимо добавить еще одну (на рисунке 19 это фигура «Преподаватель»), перейдите на вкладку *Конструктор*, раскройте меню *Добавить фигуру*, выберите *Добавить фигуру ниже*.

8. Добавьте фигуры и заполните их согласно рисунку 19.

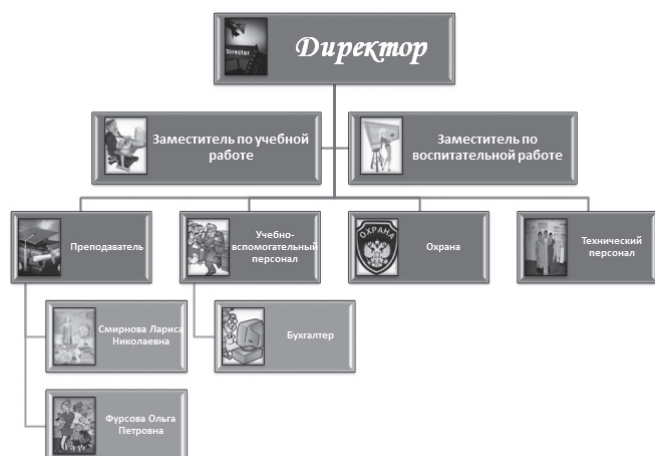


Рис. 19

Чтобы удалить фигуру, нужно выделить ее, дважды щелкнув на ней левой кнопкой мыши, и нажать клавишу Delete.

9. Растяните фигуры в длину. Для этого выделите фигуры во втором ряду (для выделения нескольких фигур следует держать нажатой клавишу Ctrl) и растяните за боковые маркеры в любом направлении до нужного размера (рис. 20).



Рис. 20

10. С помощью различных эффектов оформления, предлагаемых на вкладке **Формат**, произведите настройки построенной схемы (рис. 21).



Рис. 21

11. Для добавления анимации к диаграмме SmartArt выделите ее, перейдите на вкладку **Анимация** и в группе **Анимация** выберите эффект входа **Случайные полосы**.

12. В разделе **Параметры эффекта** выберите: **Направление** — **По вертикали**, **Последовательность** — **Последовательно по уровням** (рис. 22).



Рис. 22

Возможные варианты анимации

№ п/п	Название	Действие
1	Как один объект	Анимация применяется для всего рисунка SmartArt как для одного большого изображения
2	Все вместе	Отличие этого эффекта анимации от <i>Как один объект</i> заключается в более выраженной анимации, при которой фигуры вращаются или увеличиваются в размере: - при использовании эффекта <i>Все вместе</i> каждая фигура вращается или увеличивается в размерах отдельно от других; - при использовании эффекта <i>Как один объект</i> вращается или увеличивается в размерах весь рисунок SmartArt
3	Последовательно	Анимация применяется отдельно к каждой фигуре в порядке очереди
4	Последовательно по ветвям	Анимация применяется одновременно ко всем фигурам в одной ветви. Эффект подходит к ветвям организационной диаграммы или к макету с иерархией и аналогичен эффекту <i>Последовательно</i>
5	Сразу по уровням	Ко всем фигурам одного уровня анимация применяется одновременно. Например, если в макете имеются три фигуры, содержащие текст первого уровня, и три фигуры с текстом второго уровня, анимация будет сначала применена одновременно к трем фигурам с текстом первого уровня, а затем одновременно к трем фигурам с текстом второго уровня
6	Последовательно по уровням	К фигурам рисунка SmartArt анимация сначала применяется по уровням, а затем в пределах этого уровня — по отдельности. Например, если имеется макет с четырьмя фигурами, содержащими текст первого уровня, и тремя фигурами с текстом второго уровня, анимация будет по отдельности применена сначала к каждой из четырех фигур, содержащих текст первого уровня, а затем по отдельности к каждой из трех фигур, содержащих текст второго уровня

13. Измените порядок выполнения на обратный. Для этого перейдите на вкладку **Анимация** и щелкните на кнопке **Область анимации**.

14. Правой кнопкой мыши щелкните в списке настроек анимации и в контекстном меню выберите пункт **Параметры эффектов** (рис. 23).

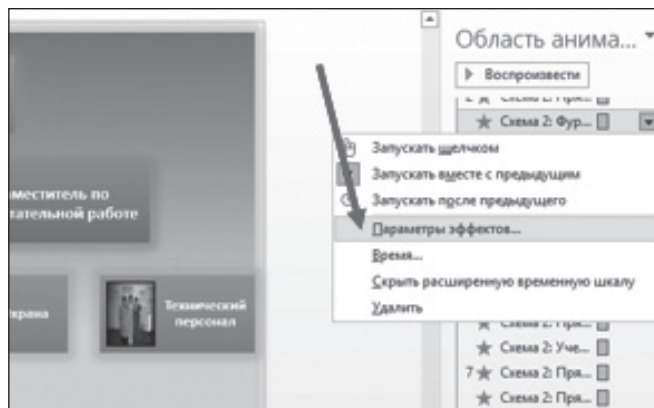


Рис. 23

15. Выберите вкладку *Анимация рисунка SmartArt* и установите флажок *В обратном порядке*. Здесь же можно изменить последовательность появления объектов диаграммы.

Домашнее задание.

Создайте следующие схемы SmartArt (рис. 24–26).
Слайд 1.



Рис. 24

Слайд 2.



Рис. 25

Слайд 3.

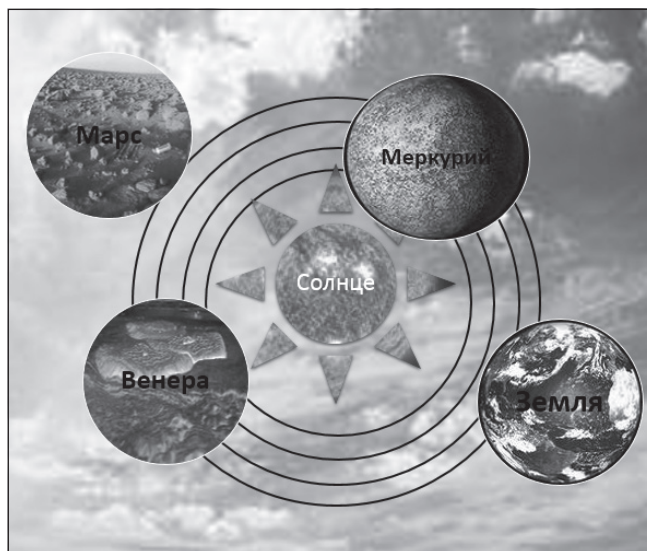


Рис. 26

Для создания диаграммы SmartArt, изображенной на слайде 3, используйте категорию *Цикл*, *Радиальная Венера* (рис. 27).

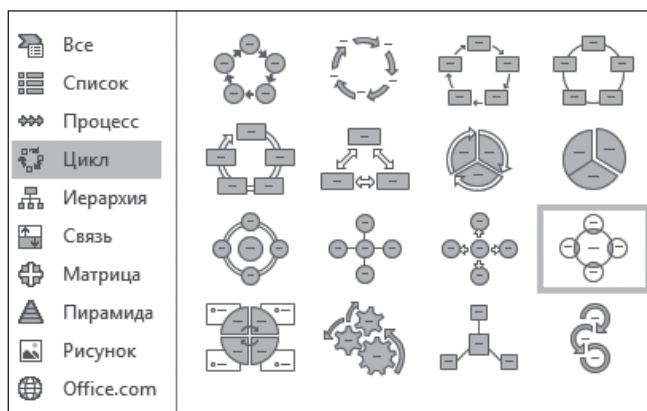


Рис. 27

Для замены формы центральной фигуры необходимо выделить ее, перейдя на вкладку *Формат* и выбрав *Изменить фигуру*.

Для всех слайдов настройте различный фон, для этого воспользуйтесь на вкладке *Дизайн* меню *Стили Фона*, *Формат фона*.

Для всех слайдов настройте эффекты перехода, для этого воспользуйтесь вкладкой *Переходы*.

Для всех созданных диаграмм SmartArt настройте различные виды анимации.

Сохраните созданную презентацию.

Список использованных источников

1. *Леонов В.* PowerPoint 2010. М.: Эксмо, 2010.
2. *Рапуто А. Г.* Визуализация как неотъемлемая составляющая процесса обучения преподавателей // III Всероссийская научно-практическая интернет-конференция «Инновационные направления в педагогическом образовании». <http://econf.gae.ru/article/5147>
3. *Шульгин В. П., Финков М. В., Прокди Р. Г.* Создание эффектных презентаций с использованием PowerPoint 2013 и других программ. М.: Наука и техника, 2015.



М. А. Плаксин,

*Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал);
Пермский государственный национальный исследовательский университет*

ПРОПЕДЕВТИКА ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ШКОЛЬНОЙ ИНФОРМАТИКЕ. МНОГОПОТОЧНОСТЬ И МНОГОЗАДАЧНОСТЬ*

Аннотация

Предложена методика знакомства учащихся с различием понятий «многопоточность» и «многозадачность» в виде пьесы. Даются сценарий, описание реквизита, теоретический материал, замечания по постановке.

Ключевые слова: информатика, средняя школа, методика обучения, многопоточность, многозадачность, параллельное программирование, пропедевтика, пьеса.

Контактная информация

Плаксин Михаил Александрович, канд. физ.-мат. наук;

доцент кафедры информационных технологий в бизнесе Пермского филиала Национального исследовательского университета «Высшая школа экономики»; *адрес:* 614070, г. Пермь, ул. Студенческая, д. 38; *телефон:* (342) 205-52-50;

доцент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета; *адрес:* 614990, г. Пермь, ул. Букирева, д. 15; *телефон:* (342) 239-67-72;

e-mail: mapl@list.ru

M. A. Plaksin,
National Research University Higher School of Economics, Perm Branch;
Perm State National Research University

PROPAEDEUTICS OF PARALLEL COMPUTING IN SCHOOL INFORMATICS. MULTITHREADING AND MULTITASKING

Abstract

The technique of acquaintance of students with the difference in the concepts of "multithreading" and "multitasking" in the form of a play is presented. There are the script, props description, the theoretical material, comments on the production.

Keywords: informatics, secondary school, teaching methods, multithreading, multitasking, parallel programming, propaedeutics, play.

Введение

Современный этап развития computer science связан с массовым распространением параллелизма вычислений на всех уровнях (многоядерные процессоры, многомашиные кластеры, многопроцессорные ЭВМ). Это делает актуальным включение пропедевтики параллельного программирования в школьный курс информатики.

На конференции «Параллельные вычислительные технологии (ПаВТ) — 2016» (Северный (Арктический) федеральный университет, Архангельск, 28 марта — 1 апреля 2016 года) прозвучала мысль, что студентам, изучающим параллельное программирование, трудно дается понятие «поток». В данной публикации предлагается методика, нацеленная на преодоление этой трудности, а именно на понимание разницы между понятиями «поток» и «задача» («процесс»). Соответствующий материал предлагается дать учащимся в виде пьесы.

В статье представлены текст пьесы, описание реквизита, теоретический материал для учителя, замечания для постановщика пьесы.

МНОГОПОТОЧНОСТЬ И МНОГОЗАДАЧНОСТЬ Пьеса в двух действиях

ПРОЛОГ

В е д у щ и й. Сегодня мы разберем два важных понятия. Вы знаете, что современные вычислительные машины способны выполнять одновременно несколько программ, решать одновременно несколько задач. Такой способ работы называется мультипрограммированием или многозадачностью. Многозадачность бывает двух видов: поточная и процессная. При поточной многозадачности выполняется сразу несколько потоков, при процессной — несколько процессов.

Дальше вместо слов «поточная многозадачность» мы будем говорить «многопоточность», а вместо слов «процессная многозадачность» — просто «многозадачность». После просмотра пьесы вы должны будете ответить на вопрос, чем многопоточность отличается от многозадачности.

По ходу пьесы три поросенка — Наф-Наф, Ниф-Ниф и Нуф-Нуф — будут готовить обед. Обед будет состоять из трех блюд: овощного супа, вареной картошки и компота. Рецепты этих трех блюд вы видите перед собой. Суп они будут варить из пары картошек, одной морковки, одной свеклы и одного кочана капусты. Компот — из трех яблок. Вареную картош-

* Материалы к статье (в частности, листы с реквизитом и фото оформления спектакля) можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_08-2017/

ку — из трех картофелин. Каждое блюдо будет вариться в отдельной кастрюле. Считаем, что кастрюли уже стоят на плите и наполнены водой.

В реальности овощи и фрукты надо помыть, почистить и порезать. В нашей пьесе овощи и фрукты напечатаны на бумаге. Их надо будет вырезать из листа бумаги (будем считать, что таким образом мы их чистим), а потом карандашом нарисовать на них поперечные линии (будем считать, что таким образом мы их нарезаем). Такие «почищенные» и «порезанные» овощи надо «положить» в соответствующую кастрюлю, т. е. наклеить на изображение кастрюли изображение овощей.

ДЕЙСТВИЕ ПЕРВОЕ

Многопоточность

ДЕЙСТВУЮЩИЕ ЛИЦА

Наф - Наф
Ни ф - Ни ф
Ну ф - Ну ф

— потоки.

Выгородка на сцене.

Коммунальная квартира.

На задней стенке висят таблички: «Комната Наф-Нафа», «Комната Ниф-Нифа», «Комната Нуф-Нуфа». Между ними таблички с надписями «Тонкая стенка».

Кроме того, есть отдельное помещение с табличкой «Общее хранилище ресурсов». В нем висят:

- рецепт приготовления вареной картошки;
- рецепт приготовления овощного супа;
- рецепт приготовления компота;
- лист с надписью «Ящик с морковью», на котором изображена морковь;
- лист с надписью «Ящик со свеклой», на котором изображена свекла;
- лист с надписью «Ящик с капустой», на котором изображены кочаны капусты;
- лист с надписью «Ящик с яблоками», на котором изображены яблоки;
- лист с надписью «Сахар». На этом листе внизу еще раз написано слово «Сахар». (Полоска с этим словом будет отрезана при приготовлении компота и наклеена в кастрюлю в знак того, что в компот положили сахар. Как вариант, полоска может быть отрезана заранее и наклеена малярным скотчем на лист с надписью «Сахар». Первый вариант выгоднее, так как создаст дополнительный повод для борьбы за ресурс ножницы. Подробнее об этом см. дальше по действию.);
- ножницы (одни);
- карандаш (один).

В комнате Наф-Нафа висит лист с надписью «Ящик с картошкой», на котором изображены картофелины.

В каждой комнате висят листы с надписью «Соль».

На переднем плане стоит стол, на котором висит табличка «Кухня». На столе лежат три ложки, стоят три тарелки и три чашки для воды.

Перед столом стоят рядом три стула, на спинках которых висят три таблички с надписями «Кастрюля» и одна табличка с надписью «Плита».

У ведущего стоит стакан с питьевой водой (прозрачный, чтобы воду было видно), лежат упаковка настоящей соли, столовая ложка. Перед началом действия ведущий демонстрирует зрителям пакет с солью и стакан с водой. Будет хорошо, если он даст зрителям прочитать надпись на пакете, после чего демонстративно его распечатает. Аналогично можно поступить с бутылкой воды, после чего налить воду в стакан.

Наф - Наф (*кричит из своей комнаты*). Потоки! Давайте готовить обед!

Ни ф - Ни ф и Ну ф - Ну ф. Давай!

На ф - На ф. Я варю картошку, поток Ниф-Ниф — суп, поток Нуф-Нуф — компот!

Ни ф - Ни ф и Ну ф - Ну ф. Хорошо!

Все три поросенка выходят из своих комнат, идут в кладовую («Общее хранилище ресурсов»), снимают листы с рецептами, возвращаются в свои комнаты и читают рецепты.

Ни ф - Ни ф. Потоки! У кого есть ресурс картошка?

На ф - На ф. У меня.

Ни ф - Ни ф. Дай, пожалуйста.

На ф - На ф. Сколько?

Ни ф - Ни ф. Парочку.

На ф - На ф. Минуточку.

Пока Наф-Наф и Ниф-Ниф переговариваются, Нуф-Нуф идет в кладовую, снимает лист яблок и ножницы, выходит с ними на кухню и начинает вырезать яблоки.

На ф - На ф (*выходит на кухню с листом картошки*). Поток Нуф-Нуф! Мне нужен ресурс ножницы.

Ну ф - Ну ф. Мне тоже, поток Наф-Наф.

Нуф-Нуф вырезает одно яблоко, кладет ножницы, берет в кладовой карандаш и начинает яблоко «резать» поперечными линиями. Наф-Наф сразу берет ножницы и вырезает две картофелины. После чего кладет ножницы.

На ф - На ф (*дает картофелины Ниф-Нифу*). Поток Ниф-Ниф! Держи!

Ни ф - Ни ф. Спасибо, поток Наф-Наф!

Ниф-Ниф идет в кладовую, снимает листы с морковью, капустой и свеклой и выходит с ними на кухню.

Все три поросенка начинают вырезать свои овощи и фрукты, расчерчивать их карандашом и складывать каждый в свою кастрюлю. После вырезания яблок Нуф-Нуф идет в кладовку за сахаром и «добавляет» его в свою кастрюлю.

Поскольку ножницы только одни и карандаш только один, между поросенками постоянно возникают споры из-за ножниц и карандаша. Эти споры должны озвучиваться и привлекать внимание зрителей. Каждый норовит заполучить ресурс (ножницы или карандаш), как только тот освободится. Чтобы возникло больше суеты, пусть каждый поросенок вырезает за один раз по одному овощу (фрукту), после чего кладет ножницы и начинает добиваться карандаша. А раскрасив и наклеив картинку в кастрюлю, опять начинает добиваться ножниц. При этом постоянно звучат фразы: «Поток такой-то, мне нужен ресурс ножницы», «Поток такой-то, дай, пожалуйста, ресурс карандаш», «Поток такой-то, когда ты уже освободишь ресурс такой-то», «Подожди минуточку, поток такой-то». Важно, чтобы постоянно повторялись слова «поток» и «ресурс».

В конце концов на плите оказываются три кастрюли, в каждой из которых варится свое блюдо.

Все три поросенка уходят по своим комнатам.

Из своей комнаты выходит Наф-Наф. Подходит к плите. Пробует суп.

На ф - На ф. Несоленый!

Уходит в свою комнату.

Из своей комнаты выходит Ниф-Ниф. Подходит к плите. Пробует суп.

Ни ф - Ни ф. Несоленый!

Уходит в свою комнату.

Из своей комнаты выходит Нуф-Нуф. Подходит к плите. Пробует суп.

Ну ф - Ну ф. Несоленый!

Уходит в свою комнату.

Из своей комнаты выходит Наф-Наф. Он несет лист с надписью «Соль».

Наф-Наф высыпает соль в кастрюлю с супом. Соответственно, на дно кастрюли ведущий наклеивает полоску бумаги

с надписью «Соль». Затем в стакан с водой он демонстративно добавляет порцию соли и размешивает.

Наф-Наф уходит к себе в комнату.

Из своей комнаты выходит Ниф-Ниф. Он несет лист с надписью «Соль».

Ниф-Ниф высыпает соль в кастрюлю с супом. Соответственно, ведущий наклеивает вторую полоску бумаги с надписью «Соль» над первой. И снова в стакан с водой он демонстративно добавляет порцию соли и размешивает.

Ниф-Ниф уходит к себе в комнату.

Из своей комнаты выходит Нуф-Нуф. Он несет лист с надписью «Соль».

Нуф-Нуф высыпает соль в кастрюлю с супом. Соответственно, наклеивает третью полоску бумаги с надписью «Соль» над второй. Кастрюля почти полна солью! В стакан с водой ведущий демонстративно добавляет порцию соли и размешивает.

Нуф-Нуф уходит к себе в комнату.

На плите остаются три кастрюли, из которых одна почти полна солью.

Из комнаты выходит Ниф-Ниф, подходит к плите.

Н и ф - Н и ф. Суп готов!

Из своих комнат выходят Наф-Наф и Нуф-Нуф. Разливают по тарелкам суп.

Ведущий берет стакан с водой (пересоленной) и отликает в чашки по несколько глотков. Поросята пробуют пересоленную воду.

Занавес.

ДЕЙСТВИЕ ВТОРОЕ

Многозадачность

ДЕЙСТВУЮЩИЕ ЛИЦА

Наф - Наф
Н и ф - Н и ф
Нуф - Нуф

— задачи.

Посыльные для передачи сообщений — двое.

Выгородка на сцене.

На задней стенке висят таблички «Дом Наф-Нафа», «Дом Ниф-Нифа», «Дом Нуф-Нуфа». Между ними таблички с надписями «Большое расстояние». (Подразумевается, что расстояние между домами велико, поэтому переговариваться задачи не могут — они должны посылать друг другу сообщения с посыльными.)

В каждом доме висит табличка «Собственное хранилище ресурсов», под которой располагаются ресурсы соответствующего поросенка.

В доме Наф-Нафа под табличкой «Собственное хранилище ресурсов» висят рецепт приготовления вареной картошки, ножницы, карандаш, лист с надписью «Ящик с картошкой», на котором изображены картофелины.

В доме Ниф-Нифа под табличкой «Собственное хранилище ресурсов» висят рецепт приготовления овощного супа, ножницы, карандаш, листы с надписями «Ящик с морковью», «Ящик со свеклой» и «Ящик с капустой», на которых изображены морковь, свекла и капуста.

В доме Нуф-Нуфа под табличкой «Собственное хранилище ресурсов» висят рецепт приготовления компота, ножницы, карандаш, лист с надписью «Ящик с яблоками», на котором изображены яблоки, и лист с надписью «Сахар». Как и в первом действии, на листе «Сахар» заготовлена полоска с надписью «Сахар», которая будет «добавлена в кастрюлю с компотом».

Кроме того, в каждой комнате висят листы с надписью «Соль».

На переднем плане стоят три стола с табличками «Кухня» (в каждом доме кухня своя). На каждом столе находятся ложка, тарелка. Перед каждым столом стоит стул с табличками «Плита» и «Кастрюля».

На столе у каждого поросенка заготовлена пачка сообщений, которые этот поросенок будет писать по ходу пьесы. То есть реально поросенок ничего не пишет, просто читает вслух уже написанное. Адресат письма — это либо конкретная задача,

либо слово «циркулярно». Циркулярные письма разносятся всем задачам (в нашем случае их только две). Поэтому циркулярные письма заготовлены в двух экземплярах.

Наф - Наф («пишет» сообщение). Отправитель: задача Наф-Наф. Адресат: циркулярно.

Встают и подходят два посыльных.

Наф - Наф. Сообщение: «Давайте готовить обед!»
Отправить сообщение!

Посыльные берут листы с сообщениями и идут к домам двух других поросят.

Откуда взялся второй лист, если Наф-Наф писал только один раз, остается непонятным, но это и не важно. Тем паче, что Наф-Наф ничего не пишет, а просто берет со стола лист и читает заранее заготовленный текст.

Дабы подчеркнуть тот факт, что расстояние между домами велико, посыльные должны находиться в пути достаточно долго.

Посыльные (приходят к Ниф-Нифу и Нуфу-Нуфу). Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

Н и ф - Н и ф и Нуф - Нуф. Принять сообщение!

Посыльные вручают сообщения и уходят со сцены.

Ниф-Ниф и Нуф-Нуф читают сообщения, после чего начинают «писать».

Н и ф - Н и ф. Отправитель: задача Ниф-Ниф. Адресат: задача Наф-Наф.

Встает и подходит посыльный 1.

Н и ф - Н и ф. Сообщение: «Давай!» Отправить сообщение!

Посыльный 1 берет лист с сообщением и идет к Наф-Нафу.

Нуф - Нуф (одновременно с Ниф-Нифом). Отправитель: задача Нуф-Нуф. Адресат: задача Наф-Наф.

Встает и подходит посыльный 2.

Нуф - Нуф. Сообщение: «Давай!» Отправить сообщение!

Посыльный 2 берет лист с сообщением и идет к Наф-Нафу. Посыльные приходят к Наф-Нафу.

Посыльный 1. Тук-тук! Вам поступило сообщение от задачи Ниф-Ниф.

Наф - Наф. Принять сообщение!

Посыльный 2. Тук-тук! Вам поступило сообщение от задачи Нуф-Нуф.

Наф - Наф. Принять сообщение!

Посыльные вручают сообщения и уходят со сцены.

Наф - Наф («пишет» письмо). Отправитель: задача Наф-Наф. Адресат: циркулярно.

Встают и подходят два посыльных.

Наф - Наф. Сообщение: «Я варю картошку, Ниф-Ниф — суп, Нуф-Нуф — компот». Отправить сообщение!

Посыльные берут листы с сообщениями и идут к домам двух других поросят.

Посыльные (приходят к Ниф-Нифу и Нуфу-Нуфу). Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

Н и ф - Н и ф и Нуф - Нуф. Принять сообщение!

Посыльные вручают сообщения и уходят со сцены.

Ниф-Ниф и Нуф-Нуф читают сообщения, после чего начинают «писать».

Н и ф - Н и ф. Отправитель: задача Ниф-Ниф. Адресат: задача Наф-Наф.

Встает и подходит посыльный 1.

Н и ф - Н и ф. Сообщение: «Хорошо!» Отправить сообщение!

Посыльный 1 берет лист с сообщением и идет к Наф-Нафу.

Н у ф - Н у ф (*одновременно с Ниф-Нифом*). Отправитель: задача Нуф-Нуф. Адресат: задача Наф-Наф.

Встает и подходит посыльный 2.

Н у ф - Н у ф. Сообщение: «Хорошо!» Отправить сообщение!

Посыльный 2 берет лист с сообщением и идет к Наф-Нафу. Посыльные приходят к Наф-Нафу

П о с ы л ь н ы й 1. Тук-тук! Вам поступило сообщение от задачи Ниф-Ниф.

Н а ф - Н а ф. Принять сообщение!

П о с ы л ь н ы й 2. Тук-тук! Вам поступило сообщение от задачи Нуф-Нуф.

Н а ф - Н а ф. Принять сообщение!

Посыльные вручают сообщения и уходят со сцены.

Н и ф - Н и ф (*«пишет» письмо*). Отправитель: задача Ниф-Ниф. Адресат: циркулярно.

Встают и подходят два посыльных.

Н и ф - Н и ф. Сообщение: «У кого есть ресурс картошка?» Отправить сообщение!

Посыльные берут листы с сообщениями и идут к домам двух других поросят.

П о с ы л ь н ы е (*приходят к Наф-Нафу и Нуф-Нуфу*). Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

Н а ф - Н а ф и Н у ф - Н у ф. Принять сообщение!

Посыльные вручают сообщения и уходят со сцены.

Наф-Наф и Нуф-Нуф читают сообщения.

Затем Наф-Наф начинает «писать».

Н а ф - Н а ф. Отправитель: задача Наф-Наф. Адресат: задача Ниф-Ниф.

Встает и подходит посыльный 1.

Н а ф - Н а ф. Сообщение: «У меня». Отправить сообщение!

Посыльный 1 берет лист с сообщением и идет к Ниф-Нифу.

П о с ы л ь н ы й 1 (*приходит к Ниф-Нифу*). Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

Н и ф - Н и ф. Принять сообщение!

Посыльный 1 вручает сообщение и уходит со сцены.

Ниф-Ниф читает сообщение, после чего начинает «писать».

Н и ф - Н и ф. Отправитель: задача Ниф-Ниф. Адресат: задача Наф-Наф.

Встает и подходит посыльный 1.

Н и ф - Н и ф. Сообщение: «Дай, пожалуйста». Отправить сообщение!

Посыльный 1 берет лист с сообщением и идет к Наф-Нафу.

П о с ы л ь н ы й 1 (*приходит к Наф-Нафу*). Тук-тук! Вам поступило сообщение от задачи Ниф-Ниф.

Н а ф - Н а ф. Принять сообщение!

Посыльный 1 вручает сообщение и уходит со сцены.

Наф-Наф читает сообщение, после чего начинает «писать».

Н а ф - Н а ф. Отправитель: задача Наф-Наф. Адресат: задача Ниф-Ниф.

Встает и подходит посыльный 1.

Н а ф - Н а ф. Сообщение: «Сколько?» Отправить сообщение!

Посыльный 1 берет лист с сообщением и идет к Ниф-Нифу.

П о с ы л ь н ы й 1 (*приходит к Ниф-Нифу*). Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

Н и ф - Н и ф. Принять сообщение!

Посыльный 1 вручает сообщение и уходит со сцены.

Ниф-Ниф читает сообщение, после чего начинает «писать».

Н и ф - Н и ф. Отправитель: задача Ниф-Ниф. Адресат: задача Наф-Наф.

Встает и подходит посыльный 1.

Н и ф - Н и ф. Сообщение: «Парочку». Отправить сообщение!

Посыльный 1 берет лист с сообщением и идет к Наф-Нафу.

П о с ы л ь н ы й 1 (*приходит к Наф-Нафу*). Тук-тук! Вам поступило сообщение от задачи Ниф-Ниф.

Н а ф - Н а ф. Принять сообщение!

Посыльный 1 вручает сообщение и уходит со сцены.

Наф-Наф читает сообщение, после чего берет ножницы, вырезает две картофелины.

Н а ф - Н а ф. Отправитель: задача Наф-Наф. Адресат: задача Ниф-Ниф.

Встает и подходит посыльный 1.

Н а ф - Н а ф (*дает посыльному 1 лист с пустым сообщением и две картофелины*). Отправить сообщение!

Посыльный 1 берет лист с пустым сообщением и картофелины и идет к Ниф-Нифу.

П о с ы л ь н ы й 1 (*приходит к Ниф-Нифу*). Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

Н и ф - Н и ф. Принять сообщение!

Посыльный 1 вручает лист с сообщением и две картофелины и уходит со сцены.

Каждый поросенок в своем доме вырезает свои овощи/фрукты, разлиновывает их карандашом, кладет их в кастрюлю. Нуф-Нуф «добавляет» в компот сахар.

Ниф-Ниф «добавляет» в суп соль (ведущий на кастрюлю наклеивает полоску «Соль»).

Н и ф - Н и ф (*«пишет» письмо*). Отправитель: задача Ниф-Ниф. Адресат: циркулярно.

Встают и подходят два посыльных.

Н и ф - Н и ф. Сообщение: «Суп готов!» Отправить сообщение!

Посыльные берут листы с сообщениями и идут к домам двух других поросят.

П о с ы л ь н ы е (*приходят к Наф-Нафу и Нуф-Нуфу*). Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

Н а ф - Н а ф и Н у ф - Н у ф. Принять сообщение!

Посыльные вручают сообщения и уходят со сцены.

Наф-Наф и Нуф-Нуф читают сообщения. После чего начинают «писать».

Н а ф - Н а ф. Отправитель: задача Наф-Наф. Адресат: задача Ниф-Ниф.

Встает и подходит посыльный 1.

На ф - На ф. Сообщение: «Пришли, пожалуйста, мою порцию».

Наф-Наф дает посыльному лист с сообщением и тарелку.

На ф - На ф. Отправить сообщение!

Посыльный 1 берет лист с сообщением и тарелку и идет к Ниф-Нифу.

Ну ф - Ну ф. Отправитель: задача Нуф-Нуф. Адресат: задача Ниф-Ниф.

Встает и подходит посыльный 2.

Ну ф - Ну ф. Сообщение: «Пришли, пожалуйста, мою порцию».

Нуф-Нуф дает посыльному 2 лист с сообщением и тарелку.

Ну ф - Ну ф. Отправить сообщение!

Посыльный 2 берет лист с сообщением и тарелку и идет к Ниф-Нифу.

Посыльные приходят к Ниф-Нифу.

П о с ы л ь н ы й 1. Тук-тук! Вам поступило сообщение от задачи Наф-Наф.

П о с ы л ь н ы й 2. Тук-тук! Вам поступило сообщение от задачи Нуф-Нуф.

Н и ф - Н и ф (*обращаясь к посыльному 1*). Принять сообщение!

Посыльный 1 вручает сообщение и тарелку и уходит со сцены.

Посыльный 2 ждет.

Ниф-Ниф читает сообщение.

Берет тарелку Наф-Нафа и «наливает» в нее суп.

Затем начинает «писать».

Н и ф - Н и ф. Отправитель: задача Ниф-Ниф. Адресат: задача Наф-Наф.

Встает и подходит посыльный 1.

Ниф-Ниф вручает ему лист с пустым сообщением и тарелку.

Н и ф - Н и ф. Отправить сообщение!

Посыльный 1 берет лист с пустым сообщением и тарелку и идет к Наф-Нафу.

Н и ф - Н и ф (*обращаясь к посыльному 2*). Принять сообщение!

Посыльный 2 вручает сообщение и тарелку и уходит со сцены.

Ниф-Ниф читает сообщение.

Ниф-Ниф берет тарелку Нуф-Нуфа и «наливает» в нее суп. Затем начинает «писать».

Н и ф - Н и ф. Отправитель: задача Ниф-Ниф. Адресат: задача Нуф-Нуф.

Встает и подходит посыльный 2.

Ниф-Ниф вручает ему лист с пустым сообщением и тарелку.

Н и ф - Н и ф. Отправить сообщение!

Посыльный 2 берет лист с пустым сообщением и тарелку и идет к Нуф-Нуфу.

В то время как Ниф-Ниф общался с посыльным 2, наливал и отправлял суп Нуф-Нуфу, посыльный 1 несет суп Наф-Нафу.

П о с ы л ь н ы й 1 (*приходит к Наф-Нафу*). Тук-тук! Вам поступило сообщение от задачи Ниф-Ниф.

На ф - На ф. Принять сообщение!

Наф-Наф берет лист с сообщением и тарелку с супом и начинает «есть» суп. Посыльный 1 уходит со сцены.

П о с ы л ь н ы й 2 (*приходит к Нуф-Нуфу*). Тук-тук! Вам поступило сообщение от задачи Ниф-Ниф.

Ну ф - Ну ф. Принять сообщение!

Нуф-Нуф берет лист с сообщением и тарелку с супом и начинает «есть» суп. Посыльный 2 уходит со сцены.

Ниф-Ниф «наливает» себе суп и тоже начинает есть.

Занавес.

* * *

Ведущий (учитель) проводит обсуждение спектакля. Учащиеся должны ответить на вопрос, поставленный в самом начале: чем отличаются потоки от процессов (задач), многопоточность от многозадачности? В результате обсуждения надо прийти к следующим выводам.

Потоки все вместе работают с одними и теми же ресурсами. Все ресурсы находятся в одном «общем хранилище». Любой поток может в любой момент обратиться к любому ресурсу. Никакой защиты нет. Даже если есть договоренность о разделении ресурсов, любой поток может эту договоренность нарушить, и никто не сможет ему помешать. Такое совместное использование ресурсов может привести к ошибкам (трижды посоленный суп).

Для понимания потоков можно использовать метафору коммунальной квартиры.

У процессов ресурсы у каждого свои. Обратиться напрямую к чужим ресурсам (ресурсам другого процесса) процесс не может. Если процессу надо получить что-либо из чужих ресурсов, он должен обратиться за разрешением к владельцу этих ресурсов. И уже владелец решает, разрешить доступ или нет. В данном случае доступ к ресурсам строго контролируется владельцами этих ресурсов.

Для понимания процессов можно использовать метафору отдельных квартир.

Совместное использование одних и тех же ресурсов несколькими потоками, с одной стороны, создает конкуренцию между потоками за право доступа к ресурсу (поток вынужден ждать, пока другой поток освободит ресурс), с другой — позволяет обойтись минимальным количеством ресурсов. Каждый ресурс будет загружен в максимальной степени, будет использоваться всеми потоками, которым он нужен.

Тот факт, что каждый процесс полностью распоряжается своими ресурсами, с одной стороны, обеспечивает каждому процессу мгновенный доступ к любому из имеющихся у него ресурсов (ждать никого и ничего не надо), с другой — требует дублирования ресурсов. Если несколько процессов нуждаются в одном и том же ресурсе, каждый процесс будет претендовать на свой экземпляр данного ресурса. Ресурсы, не нужные в данный момент их владельцу, будут простаивать.

Взаимодействие между потоками организуется легко и быстро (как общение соседей в коммунальной квартире). Взаимодействие между процессами требует специальных механизмов, занимает много сил и времени (как посылка писем по почте или передача их с посыльным).

При отправке сообщений адресат «циркулярно» означает, что сообщение должно быть послано всем другим процессам сразу.

Кроме того, в ходе обсуждения полезно будет использовать теоретический материал, изложенный ниже в разделе «О некоторых терминах, связанных с мультипрограммированием (пояснения для учителя)».

Реквизит

Ножницы — 4 штуки. Ножницы должны быть небольшими, чтобы их можно было приклеить к стене малярным скотчем. Одновременно задействованы только трое ножниц. Но лучше иметь четыре: одни для первого действия и три — для второго.

Карандаш — 4 штуки. Карандаши тоже надо будет приклеивать к стене. Количество — аналогично количеству ножниц.

Ложки (одноразовые) — 3 штуки.

Тарелки (одноразовые) — 3 штуки.

Стакан с питьевой водой (прозрачный) — 1 штука.

Соль пищевая — 1 упаковка.

Ложка столовая — 1 штука.

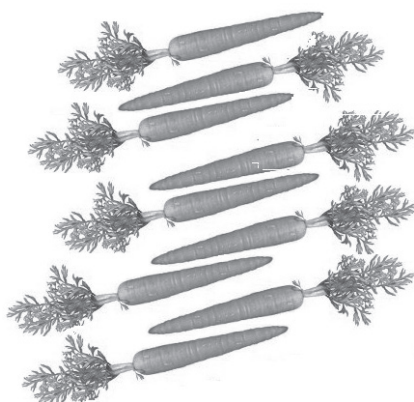
Чашки (одноразовые) для соленой воды — 3 штуки.

Листы с изображениями овощей и фруктов — по 2 штуки (по одному для каждого действия пьесы):

- с надписью «Ящик с картошкой» и изображениями картофеля:



- с надписью «Ящик с морковью» и изображениями морковок:



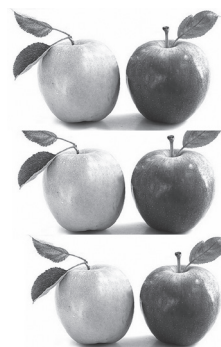
- с надписью «Ящик со свеклой» и изображениями свеклы:



- с надписью «Ящик с капустой» и изображениями кочанов капусты:



- с надписью «Ящик с яблоками» и изображениями яблок:



Листы с надписями:

- «Комната Наф-Нафа»;
- «Комната Ниф-Нифа»;
- «Комната Нуф-Нуфа»;
- «Тонкая стенка» — 2 штуки;
- «Общее хранилище ресурсов»;
- «Дом Наф-Нафа»;
- «Дом Ниф-Нифа»;
- «Дом Нуф-Нуфа»;
- «Большое расстояние» — 2 штуки;
- «Собственное хранилище ресурсов» — 3 штуки;
- «Кухня» — 3 штуки;
- «Плита» — 3 штуки;
- «Кастрюля» — 3 штуки (лучше 6 штук);
- «Соль» — 4 штуки.
- «Сахар» — 2 штуки.

Листы с рецептом каждого блюда — по 2 штуки:

- «Рецепт супа: 1 кочан капусты, 2 картофелины, 1 свекла, 1 морковь»;
- «Рецепт вареной картошки: 3 картофелины»;
- «Рецепт компота: 3 яблока, сахар».

Три полоски с надписью «Соль», которые будут наклеиваться на «кастрюлю с супом». Полоски должны быть шириной с «кастрюлю» и высотой примерно в четверть «кастрюли», таким образом, чтобы после их наклейки «кастрюля» была почти заполнена «солью».

Две полоски с надписью «Сахар».

Ниже перечислены карточки с сообщениями, которые посылают задачи.

Циркулярные сообщения заготовлены в двух экземплярах.

Для каждого сообщения указывается его сквозной номер и номер в списке сообщений данного поросенка.

1. (Наф-Наф — 1) Отправитель: задача Наф-Наф Адресат: циркулярно Сообщение: «Давайте готовить обед!»
1. (Наф-Наф — 1) Отправитель: задача Наф-Наф Адресат: циркулярно Сообщение: «Давайте готовить обед!»
2. (Ниф-Ниф — 1) Отправитель: задача Ниф-Ниф Адресат: задача Наф-Наф Сообщение: «Давай!»
3. (Нуф-Нуф — 1) Отправитель: задача Нуф-Нуф Адресат: задача Наф-Наф Сообщение: «Давай!»
4. (Наф-Наф — 2) Отправитель: задача Наф-Наф Адресат: циркулярно Сообщение: «Я варю картошку, Ниф-Ниф — суп, Нуф-Нуф — компот»
4. (Наф-Наф — 2) Отправитель: задача Наф-Наф Адресат: циркулярно Сообщение: «Я варю картошку, Ниф-Ниф — суп, Нуф-Нуф — компот»
5. (Ниф-Ниф — 2) Отправитель: задача Ниф-Ниф Адресат: задача Наф-Наф Сообщение: «Хорошо!»
6. (Нуф-Нуф — 2) Отправитель: задача Нуф-Нуф Адресат: задача Наф-Наф Сообщение: «Хорошо!»
7. (Ниф-Ниф — 3) Отправитель: задача Ниф-Ниф Адресат: циркулярно Сообщение: «У кого есть ресурс картошка?»
7. (Ниф-Ниф — 3) Отправитель: задача Ниф-Ниф Адресат: циркулярно Сообщение: «У кого есть ресурс картошка?»
8. (Наф-Наф — 3) Отправитель: задача Наф-Наф Адресат: задача Ниф-Ниф Сообщение: «У меня»
9. (Ниф-Ниф — 4) Отправитель: задача Ниф-Ниф Адресат: задача Наф-Наф Сообщение: «Дай, пожалуйста»
10. (Наф-Наф — 4) Отправитель: задача Наф-Наф Адресат: задача Ниф-Ниф Сообщение: «Сколько?»
11. (Ниф-Ниф — 5) Отправитель: задача Ниф-Ниф Адресат: задача Наф-Наф Сообщение: «Парочку»
12. (Наф-Наф — 5) Отправитель: задача Наф-Наф. Адресат: задача Ниф-Ниф. { Для передачи картошки! В роли сообщения выступают пересылаемые картофелины. }
13. (Ниф-Ниф — 6) Отправитель: задача Ниф-Ниф Адресат: циркулярно Сообщение: «Суп готов!»
13. (Ниф-Ниф — 6) Отправитель: задача Ниф-Ниф Адресат: циркулярно Сообщение: «Суп готов!»
14. (Наф-Наф — 6) Отправитель: задача Наф-Наф Адресат: задача Ниф-Ниф Сообщение: «Пришли, пожалуйста, мою порцию» { Передать посыльному тарелку. }
15. (Нуф-Нуф — 3) Отправитель: задача Нуф-Нуф Адресат: задача Ниф-Ниф Сообщение: «Пришли, пожалуйста, мою порцию» { Передать посыльному тарелку. }
16. (Ниф-Ниф — 7) Отправитель: задача Ниф-Ниф Адресат: задача Наф-Наф { Для передачи супа! В роли сообщения выступает пересылаемая тарелка супа. }
17. (Ниф-Ниф — 8) Отправитель: задача Ниф-Ниф Адресат: задача Нуф-Нуф { Для передачи супа! В роли сообщения выступает пересылаемая тарелка супа. }

О некоторых терминах, связанных с мультипрограммированием (пояснения для учителя)

Чтобы понять, что такое мультипрограммирование (многопрограммный режим), необходимо разобраться в терминологии. Путаницы здесь много. Связана она частично с эволюцией computer science и сопутствующим ей развитием понятийного аппарата, частично с переносом терминов в русский язык из английского, частично с множеством независимых разработок, авторы которых не всегда могли и/или хотели согласовывать используемую терминологию.

Попробуем навести некоторый порядок.

Мультипрограммирование, или **многопрограммный режим**, — это способность ЭВМ выполнять сразу (параллельно) несколько программ. Параллельность может быть реальной или мнимой (псевдопараллельностью). Реальной, если имеющаяся аппаратура позволяет выполнять сразу несколько фрагментов программного кода. Мнимой, если в каждый конкретный момент времени выполняется только одна программа. Иллюзия одновременности создается за счет того, что ЭВМ регулярно переключается с выполнения одной программы на выполнение другой. При достаточно большой скорости машины и достаточно медленной работе пользователя у последнего может создаться иллюзия, что компьютер работает с ним непрерывно. Например, если человек просто набирает текст в текстовом редакторе, то темп его работы может позволить ЭВМ выполнять параллельно и другие программы так быстро, что человек этого даже не почувствует. Однако, если машина будет тратить на параллельные программы слишком много времени, пользователь это заметит.

Мультипрограммирование может быть организовано на двух уровнях — на уровне операционной системы и на уровне одной программы. Мультипрограммирование на уровне операционной системы означает, что ОС обеспечивает одновременное выполнение нескольких независимых программ, например, общение с несколькими пользователями через терминалы и запуск нужных им программ по их указанию. Мультипрограммирование на уровне одной программы означает, что программист выделяет в программе части, которые могут исполняться параллельно, и вычислительная система должна обеспечить параллельное (или псевдопараллельное) выполнение этих частей.

Многопользовательский режим — это способность ЭВМ одновременно выполнять программы, запущенные разными пользователями. Например, несколько человек могут работать с одной машиной с разных терминалов. Естественно, многопользовательский режим обязательно является многопрограммным. Но многопрограммный режим может и не быть многопользовательским. Существенное отличие многопользовательского мультипрограммирования от однопользовательского — требование надежной изоляции разных пользователей друг от друга. Нехорошо, если человек на одном терминале вводит конфиденциальную информацию, а другой человек с другого терминала может эту информацию подсмотреть. Другое дело, если один пользователь сам работает одновременно с несколькими программами. Например, сочиняет отчет в Word, формирует для него красивую диаграмму в Excel и вставляет ее в файл Word.

Практически одновременно с термином «мультипрограммирование» в computer science возник термин «мультизадачность». Понятие *«задача»* означало «программа, выполняемая параллельно с другими программами». Так, знаменитая когда-то операционная система OS/360, созданная во второй половине 1960-х годов для семейства мэйнфреймов фирмы IBM (советский аналог назывался ОС ЕС), имела версии MFT и MVT — «multiprogramming with a fixed number of tasks» (мультипрограммирование с фиксированным количеством задач) и «multiprogramming with a variable number of tasks» (мультипрограммирование с изменяемым количеством задач). В языке программирования PL/I, который разрабатывался в рамках проекта 360 и должен был стать основным языком программирования фирмы IBM, параллельно выполняющиеся подпрограммы помечались описателем TASK (задача). Эта же терминология сохранилась в языке Ада, специально разработанном во второй половине 70-х годов XX века по заказу министерства обороны США в качестве основного языка встроенных систем, призванного заменить около 400 языков, используемых в то время программистами Пентагона. В Аде программные сегменты для параллельного программирования также имеют описатель TASK.

Другим термином, который в обязательном порядке используется в операционных системах, стал термин *«процесс»*. Он применяется для уточнения понятия «программа». Термин «программа» — многозначен: им можно обозначить и текст программы, и файл с кодом программы, и программу во время выполнения ее компьютером. «Процесс» обозначал не просто программу, а «программу, выполняемую в данный момент вычислительной системой». Первоначально именно для процесса операционная система выделяла ресурсы ЭВМ: память, время процессора, файлы данных, внешние устройства. Процесс являлся одновременно «держателем ресурсов» вычислительной системы и «исполняемой единицей».

Понятие «процесс» хорошо подходит для описания мультипрограммирования на уровне операционной системы. Хуже дело обстоит с мультипрограммированием внутри одной программы. «Внутреннее» мультипрограммирование требует разделить понятия «держателя ресурсов» и «исполнимой единицы»: внутри одного процесса может оказаться несколько параллельно выполняющихся подпроцессов. Для обозначения таких подпроцессов был предложен термин *«потоки»*. (Здесь в русском языке идет расхождение с английским. В английском языке термин «поток» — stream — используется для описания обмена данными. Для описания параллельных подпроцессов применяется термин «threads» — «нити».) Любой процесс при запуске порождает «первичный» поток. Если в программе внутреннее распараллеливание отсутствует (т. е. речь идет об обычном последовательном алгоритме), тем дело и ограничивается. Если программа рассчитана на внутреннее распараллеливание, то по ходу выполнения первичного потока могут быть порождены «вторичные» потоки. Именно между потоками ОС распределяет время центрального процессора. Процесс продолжает оставаться «держателем» всех остальных ресурсов. Роль исполнимых единиц переходит к потокам.

Многопоточное программирование стало особенно востребованным после появления многоядерных процессоров. Разделение программы на потоки позволяет задействовать ресурс многоядерности и может существенно ускорить выполнение процесса. Отметим, что при отсутствии потоков (т. е. при выполнении на много-

ядерном процессоре программ, не содержащих внутреннего распараллеливания) время выполнения программы на многоядерном процессоре может оказаться больше, чем на одноядерном. Причина в том, что частота каждого ядра в многоядерном процессоре может оказаться ниже, чем частота аналогичного одноядерного процессора.

Введение термина «поток» привело к разделению многозадачности на «поточную» и «процессную». При этом термин «многопоточность» используется широко. А вот термин «многопроцессность» не прижился. Во-первых, он труднопроизносим. Во-вторых, он почти неизбежно порождает путаницу с «многопроцессорностью».

В Windows некоторые из перечисленных понятий легко «увидеть воочию». Во-первых, список открытых пользователем задач отображается на панели *задач*. Во-вторых, нажав комбинацию клавиш Ctrl+Alt+Del, вы сможете запустить Диспетчер *задач*, а в нем посмотреть вкладки Приложения, *Процессы*, Пользователи. Интересно запустить несколько экземпляров одной программы (например, Word или интернет-браузера) и посмотреть, как это отразится в перечне процессов.

Все вышесказанное не означает, что в конкретных мультипрограммных системах и текстах, посвященных мультипрограммированию, все упомянутые термины будут использоваться именно в описанном смысле.

В предлагаемой пьесе противопоставляются потоки, которые действуют на общем наборе ресурсов, и задачи, каждая из которых имеет свой собственный набор ресурсов и которые могут общаться друг с другом только с помощью специальных механизмов (в пьесе — с помощью посылки сообщений).

Замечания по постановке пьесы

Ко времени подготовки данной публикации пьеса была исполнена трижды — на уроке в пятом классе лицея № 10 г. Перми и в двух сменах Школы по параллельным вычислениям для учащихся десятых классов, которая прошла в июне 2017 года в Пермском филиале Высшей школы экономики. (Данный материал не входит в программу пятого класса. Но автору нетерпелось попробовать. Основная мысль — потоки выполняются все вместе, используя общие ресурсы; задачи выполняются каждая сама по себе и общаются друг с другом с помощью сообщений — была понятна и пятиклассникам.)

Трудности постановки связаны со сложным оформлением. Нужны многоэлементные выгородки, реквизит, расклейка большого числа надписей. Перед спектаклем это требует длительной подготовки. В перерыве между двумя действиями всё должно быть очень быстро перестроено. После спектакля нужно прибрать уцелевшие артефакты и выбросить более непригодные. Все это может потребовать изрядного времени. Как следствие, длинные паузы расхолаживают учеников, а в конце урока не остается времени на обсуждение.

По опыту трех спектаклей автор предлагает организовать оформление спектакля следующим образом (фото 1, 2).

Большинство маленьких надписей делалось на листах формата А4. Для их крепления использовался малярный скотч. Он держит не так крепко, как канцелярский, зато в отличие от канцелярского малярный, во-первых, не липнет сразу же к первой попавшейся поверхности, во-вторых, может быть приклеен-отклеен несколько раз (при известной сноровке).



Фото 1. Оформление первого действия

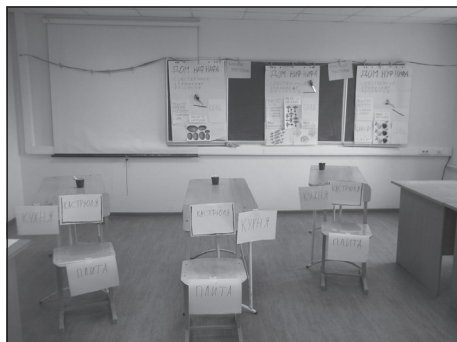


Фото 2. Оформление второго действия

Для ускорения монтажа сцены многочисленные маленькие наклейки были (в значительной степени) сгруппированы на больших листах флип-чарта.

«Общее хранилище ресурсов» показано на фото 3. Оно представляет собой два флип-чартовых листа, расположенных горизонтально и склеенных по обратной стороне канцелярским скотчем. Двух листов хватило, чтобы разместить три рецепта, названия четырех «ящиков» (моркови, свеклы, капусты и яблок) и сами эти «ящики», три «коробки с солью» и одну — с «сахаром», карандаш и ножницы. Ножницы достаточно тяжелы. Поэтому, чтобы они не отклеились, их полезно хотя бы наполовину обмотать скотчем. Названия ящиков записаны на листочках примерно в одну треть от А4. Еще один «ящик» — с картошкой — размещается не в общем хранилище, а в комнате Наф-Нафа.

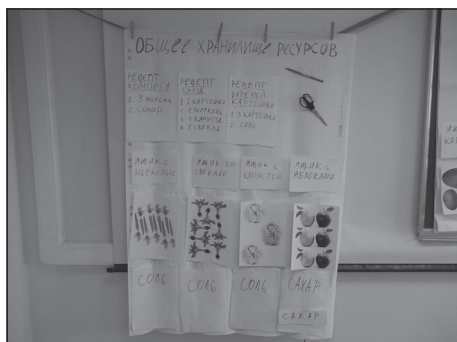


Фото 3. Общее хранилище ресурсов

Оформление дома каждого поросенка размещалось на одном вертикально расположенном листе флип-чарта (фото 4–7). Сверху — надпись: «Дом Наф-Нафа» (Ниф-Нифа, Нуф-Нуфа). Ниже — надпись: «Собственное хранилище ресурсов». Еще ниже — содержимое хранилища. Оно свое для каждого поросенка. В доме Наф-Нафа это

рецепт вареной картошки, ящик с картошкой (название и сами картофелины), соль, ножницы, карандаш. В доме Ниф-Нифа это рецепт овощного супа, ящики с морковью, свеклой и капустой, соль, ножницы, карандаш. В доме Нуф-Нуфа — рецепт компота, ящики с яблоками, сахар, соль, ножницы, карандаш. Изготовление «домовых» флип-чартовых листов рекомендуется начинать с «дома Наф-Нифа». Там требуется разместить максимально количество артефактов.

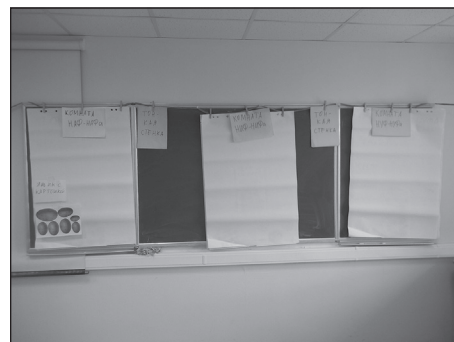


Фото 4. «Комнаты» поросят

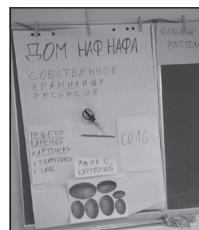


Фото 5.
Дом Наф-Нафа

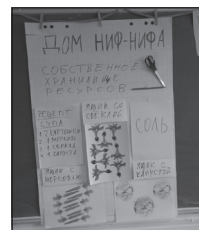


Фото 6.
Дом Ниф-Нифа

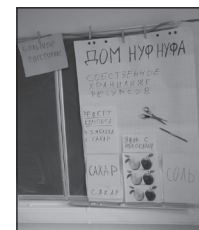


Фото 7.
Дом Нуф-Нуфа

Как уже было сказано, для крепления большинства маленьких листочков активно использовался малярный скотч. Но для «крупных единиц» (прежде всего листов флип-чарта) крепление было устроено по-другому. Вдоль верхней кромки доски, влево и вправо от нее был натянут шнур, приклеенный с помощью канцелярского скотча. Сила крепления канцелярского скотча весьма велика (автор сталкивался с ситуацией, когда скотч «официально», «по технологии» использовался для соединения частей теплиц, изготовленных заводским образом; и такие теплицы стояли по несколько лет, несмотря на превратности уральского климата). Но клеить скотч к поверхности доски не рекомендуется. Писать после этого будет сложно.

Все флип-чартовые листы были подвешены к шнуру на бельевые прищепки. Во-первых, такое крепление быстрее, чем скотч. Во-вторых, крепление прищепками — многократное. Прикрепить-открепить любой лист можно сколько угодно раз. Прищепок должно быть не меньше трех. Это позволяет манипулировать одной прищепкой, в то время как лист висит на двух других.

Для ускорения перехода от первого действия ко второму перед спектаклем были вывешены сразу все флип-чартовые листы — как с общим хранилищем ресурсов, так и с домами поросят. На время первого действия листы с домами были завешаны чистыми флип-чартовыми листами (см. фото 4). При переходе ко второму действию эти чистые листы были просто сняты. (Была идея не занавешивать листы-дома, а повернуть их к зрителю тыльной стороной. Но оказалось, что фломастерная надпись сквозь флип-чартовый лист просвечивает.)

На листах-занавесках можно сделать надписи: «Комната Наф-Нафа», «Комната Ниф-Нифа», «Комната Нуф-Нуфа». Маленькие листы с этими надписями станут не нужны.

Точно так же сразу по два перед началом спектакля были вывешены пары листов «Тонкая стенка» и «Большое расстояние».

Листы с надписью «Тонкая стенка» ориентированы вертикально, а с надписью «Большое расстояние» — горизонтально. Это затрудняет замену одного листа другим, но визуально усиливает смысл надписи.

В первом действии часть артефактов можно разместить не в общем хранилище, а в комнатах поросят, например рецепты.

Чтобы быстро приклеить вырезанные овощи к листам с надписью «Кастрюля», следует заранее нарезать узкие полоски малярного скотча и наклеить их на край «кухонного» стола.

Для ускорения работы можно вырезанные овощи не наклеивать на листы с надписью «Кастрюля», а просто складывать на стул. Еще вариант: на стулья поставить пластиковые прозрачные контейнеры, в которые и будут складываться «овощи». Однако полностью такие контейнеры листы с надписью «Кастрюля» не заменят. Важным моментом первого действия является трехкратное соление супа. Причем изобразить его последствия надо таким образом, чтобы излишнее количество соли бросалось в глаза. Для «кастрюль-листов» это сделать просто — заклеить их полосками «соли». Как это сделать для кастрюль-контейнеров — непонятно.

Листочки с сообщениями, которыми обмениваются поросята во втором действии, должны быть заранее упорядочены: сгруппированы по авторам и разложены в порядке написания.

Поскольку при переходе от первого действия ко второму придется снимать со шнура листы оформления первого действия (как большие флип-чартовские — «Общее хранилище ресурсов», листы-занавески «поросячьих домов», так и маленькие — «Комната такая-то», «Тонкая стенка»), необходимо заранее заготовить достаточно большой пустой стол, на который все эти листы будут сложены на время исполнения второго действия. Удобно, если он будет стоять с той же стороны сцены, где находится «Общее хранилище ресурсов».

Для перехода ко второму действию должны быть заготовлены два «кухонных стола». Пока они еще пусты, их можно использовать для временного складирования листов оформления

Список действий при переходе ко второму действию спектакля выглядит так:

1. Снять со стены «Общее хранилище ресурсов». Положить на заготовленный «уборочный» стол.
2. С «кухонного стола» из первого действия убрать весь мусор, обрезки, листочки, ножницы, карандаш. Сложить на «уборочный» стол.
3. Переставить на новое место «кухонный» стол из первого действия. Выдвинуть еще два «кухонных стола».
4. Подставить к столам «стулья-плиты».
5. Развесить на столы и стулья листы «Кухня», «Плита», «Кастрюля».
6. Снять надписи «Комната Наф-Нафа», «Комната Ниф-Нифа» и «Комната Нуф-Нуфа». По ходу дела складывать их на ближайший «кухонный» стол.

7. Снять надписи — листы-занавески «поросячьих домов». По ходу дела складывать их на ближайший «кухонный» стол.

8. Снять надписи «Тонкая стенка». Сложить их на ближайший «кухонный» стол. Повернуть горизонтально скрытые под ними надписи «Большое расстояние».

9. Убрать все снятые листы на «уборочный стол».

10. Расставить на «кухонные» столы одноразовую посуду.

Имеет смысл попробовать сыграть пьесу, отведя на это не один академический час, а пару часов в расчете на то, что тогда должно остаться достаточно времени для обсуждения теоретических вопросов.

В идеале хотелось бы, чтобы актеры знали свои роли наизусть и провели полноценную репетицию. Однако это вряд ли возможно. Во-первых, это требует времени. Во-вторых, полноценная репетиция означает «уничтожение» заготовленной выгородки и реквизита. И учителю придется их заново восстанавливать. Более реально, чтобы актеры заранее ознакомились со своими ролями, прочитали совместно текст по ролям и прошли свою сцену «ногами» (отрепетировали все свои перемещения по сценической площадке и обозначили выполняемые действия). В частности, полезно заготовить дополнительные листочки с «овощами», которые актеры смогут изрезать о время репетиции, не снимая такие же листочки, расклеенные для спектакля.

В заключение — две фотографии: финал первого действия спектакля и сцена из второго (фото 8, 9).

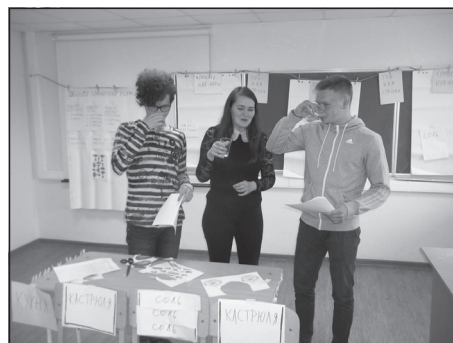


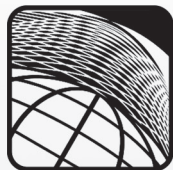
Фото 8. Финал первого действия:
опробование пересоленной воды



Фото 9. Сцена из второго действия:
Наф-Наф отправляет циркулярное сообщение

* * *

Автор будет признателен, если учителя, которые воспользуются предложенной методикой, поделятся полученным опытом.



А. А. Оленев, В. В. Малиатаки,
Ставропольский государственный педагогический институт

МОДЕЛИРОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ И ПРОСТЕЙШИХ УЗЛОВ ЭВМ В СИСТЕМЕ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ MAPLE

Аннотация

В статье рассматриваются особенности использования системы компьютерной математики Maple в учебном процессе на примере изучения тем, связанных с логическими основами построения ЭВМ и с порядком использования логических схем при таком построении.

Ключевые слова: системы компьютерной математики, Maple, компьютерное моделирование, моделирование логических элементов.

Контактная информация

Оленев Александр Анатольевич,
канд. тех. наук, доцент, доцент кафедры математики и информатики Ставропольского государственного педагогического института; *адрес:* 355029, г. Ставрополь, ул. Ленина, д. 417 «А»; *телефон:* (8652) 56-08-21; *e-mail:* olenevalexandr@gmail.com

Малиатаки Виктория Викторовна,
канд. пед. наук, ст. преподаватель кафедры математики и информатики Ставропольского государственного педагогического института; *адрес:* 355029, г. Ставрополь, ул. Ленина, д. 417 «А»; *телефон:* (8652) 56-08-21; *e-mail:* maliataki@yandex.ru

A. A. Olenov, V. V. Maliataki,
Stavropol State Pedagogical Institute

MODELING LOGICAL ELEMENTS AND THE ELEMENTARY COMPUTER ELEMENTS IN MAPLE

Abstract

The article discusses the features of using the computer mathematics system Maple in the educational process by the example of the study of topics related to the logical framework of building a computer and with the order of the use of logic schemes in this building.

Keywords: computer mathematic system, Maple, computer simulation, modeling of logical elements.

При изучении курсов информационной направленности в старших классах школы и вузах учебное время делится примерно поровну между теоретическими и практическими занятиями. При этом в ходе практических занятий обучающиеся чаще всего используют технические средства обучения, в первую очередь, компьютеры, что обуславливает построение таких занятий в форме компьютерного практикума. Нужно отметить, что компьютер целесообразно использовать не только при изучении технологических тем курсов, но и при изучении теоретических разделов.

Для освоения теоретических и практических тем курсов информационной направленности целесообразно и эффективно использовать в качестве основного методического подхода математическое моделирование. При этом объектами моделирования могут выступать как сами информационные процессы, входящие в информационную технологию (преобразование, прием и передача, обработка информации), так и отдельные элементы, реализующие отдельные функции преобразования информации. Основу этих процессов составляет именно математическое моделирование. При использовании математического моделирования происходит замена исходного объекта его абстрактным образом — вместо объекта используется математическая модель, а дальнейшее изучение модели выполняется с помощью реализуемых на компьютере алгоритмов и программ. Сам метод соединяет в себе достоинства, присущие как теории, так и экспериментам. При использовании математического моделирования работа будет выполняться не с самим объектом (явлением, процессом), а с его моделью, что и дает возможность относительно быстро и без существенных финансовых и временных затрат понять его основные свойства, оценить изменение его состояний в различных ситуациях. Исходя из этого можно достичь высокого обучающего эффекта при условии, если обучающиеся в ходе выполнения практических или лабораторных работ сами создают соответствующие учебные математические или иные модели на компьютере. Для этого могут использоваться как уже знакомые, так и еще неизвестные им технологии [1–4, 9]. Однако для построения хорошей математической модели целесообразно использовать и программное обеспечение соответствующего уровня. Системы компьютерной математики (СКМ), такие как Maple и Mathematica, наиболее полно соответствуют учебным целям в разработке именно математических моделей [1–5, 7].

В данной статье демонстрируются **возможности использования СКМ Maple при изучении тем, связанных с логическими основами построения ЭВМ и с порядком использования логических схем при таком построении** [5–8].

С представленным в работе элементом практикума обучающиеся могут работать как при изучении основных понятий темы, которые рассматриваются в ходе проведения теоретических занятий, так и в ходе практикума [9]. Интуитивно понятные функции Maple, пояснения и описания последо-

вательности действий позволяют учащимся выполнять работу *самостоятельно*.

Предлагаемые для исполнения задания располагаются в порядке возрастания сложности и в необходимом порядке изложения. Например, в учебнике И. Г. Семкина [9] последовательность изложения такая: основные понятия, логические величины, логические операции, логические формулы, простейшие логические схемы.

Задачами практической работы являются:

- моделирование в Maple логических элементов, в том числе с различным числом входов: инвертора, конъюнктора (элемент «И»), дизъюнктора (элемент «ИЛИ»);
- моделирование в Maple логических элементов универсального базиса, в том числе с различным числом входов: штриха Шеффера (элемент «И—НЕ»), стрелки Пирса (элемент «ИЛИ—НЕ»);
- объединение логических элементов в логические схемы в различных базисах в Maple.

Целью работы является обучение построению математических моделей логических элементов вычислительной техники, простейших комбинационных схем в СКА Maple, а также основным методам анализа работы логических схем. Обновление созданной модели или ее математическое описание при этом будет происходить автоматически: при изменении значений входных переменных будут пересчитываться выходные значения функции. В таблице 1 приведен пример такой программы.

Логический элемент «Инвертор» (возможные используемые обозначения: НЕ, not, $\neg$, $\bar{x}$). Логический элемент «НЕ» может иметь только один вход и один выход.

С помощью программы (п. 1, 2) можно получить таблицу истинности (п. 3, 4) для логического элемента «НЕ» в различных формах выходного представления (см. сноску к таблице 1).

Если задать значение переменной для инвертора (истинное или ложное), то схема (программа) среагирует и выдаст соответствующий результат. Пункты

5, 6 демонстрируют выходное значение при заданных значениях входных.

Можно сделать и так, чтобы вместо слов ИСТИНА (true) и ЛОЖЬ (false) в ячейках выводились 1 и 0 (демонстрация п. 4, 6).

Используя эту технологию, можно строить и анализировать работу простейших логических схем.

Для освоения данной технологии и создания библиотеки элементов основного базиса необходимо выполнить несколько заданий.

Задание 1. Построение логического элемента «НЕ».

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «НЕ» в двух вариантах (первый вариант — представление таблицы истинности в общем виде, второй вариант — в форме MOD2, см. сноску к таблице 1).

Получите таблицу истинности для элемента «НЕ» для двух форм представления.

Проследите за работой элемента, изменяя значение входного параметра в п. 5, 6.

Задание 2. Построение логического элемента «И».

Логический элемент «Конъюнктор» (возможные используемые обозначения: И, and, &, $\wedge$).

Простейший логический элемент «И» должен иметь не менее двух входов и один выход.

Программа для моделирования элемента «И» представлена в таблице 2.

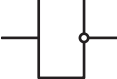
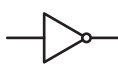
Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от выбранной (заданной) формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «И» в двух вариантах.

Получите таблицу истинности для элемента «И» для двух форм представления, используя функцию Maple TruthTable.

Таблица 1

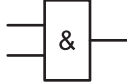
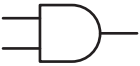
Математическое моделирование элемента «НЕ»

№ п/п	Математическое моделирование	Пояснения	Условно-графическое изображение	
			IEC (ГОСТ)	ANSI
1	<pre>> restart; > with(Logic): > элемент_НЕ:=proc(a) return &not(a); end proc;</pre>	Начало нового сеанса. Подключение библиотеки. Процедура моделирования элемента «НЕ».		
2	<pre>> T1:=элемент_НЕ(x); T1:=&not(x)</pre>	Присвоение элементу «НЕ» имени T1		
3	<pre>> T11:=TruthTable(T1,[x]); T11:=table([true=false, false=true])</pre>	Построение таблицы истинности элемента «НЕ»		
4	<pre>> T12:=TruthTable(T1,[x],form=MOD2); T12:=table([0=1, 1=0])</pre>	Построение таблицы истинности элемента «НЕ» в форме MOD2*		
5	<pre>> T11[true]; false</pre>	Проверка выходного значения отклика при входном значении true		
6	<pre>> T12[0]; 1</pre>	То же в форме MOD2		

* В СКМ Maple при использовании функции TruthTable возможен ввод и вывод на экран в виде значений true и false, а при использовании form=MOD2 значения вводятся и выводятся в виде 0 и 1 соответственно.

Таблица 2

Программа моделирования элемента «И»

Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> > элемент_И:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент И должен иметь не менее двух входов."; end if; result:=a[1] &and a[2]; for i from 3 to nops([a]) do result:=result &and a[i]; end do; return result; end proc: </pre>		

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой, изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Используя приведенную выше программу, произведите моделирование элемента «И» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 3. Построение логического элемента «ИЛИ».

Логический элемент «Дизъюнктор» (возможные используемые обозначения: ИЛИ, or, 1, ∨).

Простейший логический элемент «ИЛИ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «ИЛИ» представлена в таблице 3.

Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от выбранной (заданной) формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «ИЛИ» в двух вариантах.

Получите таблицу истинности для элемента «ИЛИ» для двух форм представления, используя функцию Maple TruthTable.

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой,

изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Создайте элемент «ИЛИ» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 4. Построение логического элемента «И—НЕ».

Логический элемент «Штрих Шеффера» (возможные используемые обозначения: И—НЕ, nand, |).

Простейший логический элемент «И—НЕ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «И—НЕ» представлена в таблице 4.

Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от выбранной (заданной) формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «И—НЕ» в двух вариантах.

Получите таблицу истинности для элемента «И—НЕ» для двух форм представления, используя функцию Maple TruthTable.

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой, изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Таблица 3

Программа моделирования элемента «ИЛИ»

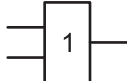
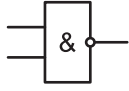
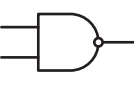
Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> > элемент_ИЛИ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент ИЛИ должен иметь не менее двух входов."; end if; result:=a[1] &or a[2]; for i from 3 to nops([a]) do result:=result &or a[i]; end do; return result; end proc: </pre>		

Таблица 4

Программа моделирования элемента «И—НЕ»

Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> > элемент_И_НЕ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент И-НЕ должен иметь не менее двух входов."; end if; result:=&not a[1] &or (&not(a[2])); for i from 3 to nops([a]) do result:=result &or (&not(a[i])); end do; return result; end proc: </pre>		

Создайте элемент «И—НЕ» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 5. Построение логического элемента «ИЛИ—НЕ».

Логический элемент «Стрелка Пирса» (возможные используемые обозначения: ИЛИ—НЕ, $\downarrow$).

Простейший логический элемент «ИЛИ—НЕ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «ИЛИ—НЕ» представлена в таблице 5.

Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «ИЛИ—НЕ» в двух вариантах.

Получите таблицу истинности для элемента «ИЛИ—НЕ» для двух форм представления, используя функцию Maple TruthTable.

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой, изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Создайте элемент «ИЛИ—НЕ» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 6. Организация связи между элементами.

Создайте элемент «И—НЕ» с помощью созданных ранее программ для моделирования элементов «И», «ИЛИ» и «НЕ».

Получите таблицы истинности и сравните с результатами (таблицами истинности) для элементов «И—НЕ» сделайте вывод.

Элемент «И» имеет два входа — x_1 и x_2 . Присвоим этому элементу имя I_2 , а элементу «НЕ» — I_3 . Выход элемента I_2 будет являться входом элемента «НЕ».

Проанализируйте полученную таблицу истинности.

Произведите сравнение с таблицей истинности, полученной для несоставного элемента «И—НЕ».

По аналогии составьте схему для элемента «НЕ—И» и получите таблицу истинности.

Сравните таблицы истинности и сделайте вывод.

Согласно классической пропозициональной логике, справедливы следующие соотношения:

$$\text{не } (x \text{ и } y) = (\text{не } x) \text{ или } (\text{не } y);$$

$$\text{не } (x \text{ или } y) = (\text{не } x) \text{ и } (\text{не } y).$$

Математически это можно представить так:

$$\neg(x \wedge y) = \neg x \vee \neg y;$$

$$\neg(x \vee y) = \neg x \wedge \neg y.$$

Самостоятельно составьте логическую схему для элемента «ИЛИ—НЕ» и выполните для него действия, идентичные предусмотренным для изучения составного элемента «И—НЕ».

Таблица 5

Программа моделирования элемента «ИЛИ—НЕ»

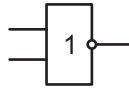
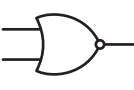
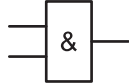
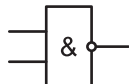
Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> > Элемент_ИЛИ_НЕ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент ИЛИ-НЕ должен иметь не менее двух входов."; end if; result:=a[1] &nor a[2]; for i from 3 to nops([a]) do result:=result &and (&not(a[i])); end do; return result; end proc: </pre>		

Таблица 6

Пример моделирования составных элементов

Математическое моделирование	Результат	Реализуемые логические элементы	
$I2 := \text{элемент_И}(x1, x2);$ $I3 := \text{элемент_НЕ}(I2);$	$I2 := x1 \text{ \& } x2$ $I3 := \text{\¬}(x1 \text{ \& } x2)$	Элемент «И»	
		Соединение элемента «И» и элемента «НЕ» — элемент «И—НЕ»	
$\text{TruthTable}(I3, [x1, x2]);$	$\text{table}([(false, false) = true,$ $(true, true) = false,$ $(false, true) = true,$ $(true, false) = true])$	Элемент «И—НЕ»	

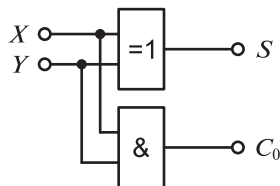
Контрольное задание.

Постройте логические схемы, соответствующие приведенным в таблице 7 логическим формулам.

Для каждой схемы проверьте правильность ее работы путем построения таблицы истинности.

Задание 7. Построение одноразрядного полусумматора.

Полусумматор реализует сложение двух двоичных слагаемых X и Y с получением на выходе младшего разряда их суммы S и старшего (переносимого) разряда C_0 .



На рисунке приведен внешний вид логической схемы полусумматора, двум выходам которого (C_0 и S) соответствуют следующие логические формулы:

$$C_0 = X \text{ \& } Y;$$

$$S = X \text{ XOR } Y.$$

Таким образом, схему одноразрядного полусумматора можно реализовать, используя два логических элемента: логического элемента «И», имеющего два входа, и логического элемента «исключающее ИЛИ».

7.1. Построение логического элемента «исключающее ИЛИ».

Логический элемент «исключающее ИЛИ» (возможные используемые обозначения: сложение (сумма) по модулю 2, исключающее_ИЛИ, неравнозначность, MOD 2, =1).

Простейший логический элемент «исключающее ИЛИ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «исключающее ИЛИ» представлена в таблице 8.

Сопоставьте полученные результаты с таблицей истинности.

7.2. Моделирование логической схемы полусумматора.

Произведите моделирование логической схемы одноразрядного полусумматора с помощью полученной ранее программы для логического элемента «И» и полученного в пункте 7.1 логического элемента «исключающее ИЛИ». Для организации связей воспользуйтесь рекомендациями задания 6.

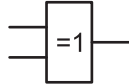
Таблица 7

Примеры использования различных обозначений логических функций

Вариант 1 (чтение)	Вариант 2
не x или y	$\neg x \text{ or } y$
x и y или z	$x \text{ and } y \text{ or } z$
x или y и z	$x \text{ or } y \text{ and } z$
не x или y и z	$\neg x \text{ or } y \text{ and } z$
не (x и y и z)	$\neg(x \text{ and } y \text{ and } z)$
(x или y) и (z или y)	$(x \text{ or } y) \text{ and } (z \text{ or } y)$
не(x или y) и (z или y)	$\neg(x \text{ and } y) \text{ or } (z \text{ or } x)$
((не x и y) или x) и не y	$((\neg x \text{ and } y) \text{ or } z) \text{ and } \neg y$

Таблица 8

Программа моделирования элемента «Исключающее ИЛИ»

Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> элемент_ИСКЛИЛИ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент ИЛИ должен иметь не менее двух входов."; end if; result:=a[1] &xor a[2]; for i from 3 to nops([a]) do result:=result &xor a[i]; end do; return result; end proc; </pre>		

Постройте таблицу истинности для выходных параметров C_0 и S полусумматора.

Протестируйте работу логической схемы, перебрав все варианты значений входных параметров X и Y .

Задание 8 (повышенной сложности). Построение одноразрядного полусумматора в универсальных базисах.

Произведите моделирование логической схемы полусумматора в универсальных базисах — штрих Шеффера и стрелка Пирса.

Постройте таблицу истинности для выходных параметров C_0 и S полусумматора в одном из универсальных базисов — штрих Шеффера и стрелка Пирса.

Протестируйте работу логической схемы, перебрав все варианты значений входных параметров X , Y , C_0 .

Сопоставьте полученные результаты с таблицей истинности.

Список использованных источников

1. Говорухин В. И., Цибулин В. Г. Введение в Maple. Математический пакет для всех. М.: Мир, 1997.
2. Дьяконов В. П. Maple 10/11/12/13/14 в математических расчетах. М.: СОЛОН-Пресс, 2011.
3. Кирсанов М. Н. Maple 13 и Maple. Решение задач механики. М.: Физматлит, 2010.

4. Красильников В. В., Оленев А. А., Тоискин В. С. Проверка знаний по теме «Множество» на основе системы компьютерной математики Maple // Актуальные вопросы инженерного образования — 2015: Сборник научных трудов международной научно-методической конференции (Октябрьский, 27 ноября 2015 г.). Екатеринбург: Универсальная Типография «Альфа Принт», 2016.

5. Малиатаки В. В., Медведева Л. М., Оленев А. А. Совершенствование математической подготовки учителя в вузе на основе использования СКА Maple // Актуальные вопросы инженерного образования — 2015: Сборник научных трудов международной научно-методической конференции (Октябрьский, 27 ноября 2015 г.). Екатеринбург: Универсальная Типография «Альфа Принт», 2016.

6. Оленев А. А. Особенности реализации алгоритма Евклида в Maple // Актуальные вопросы инженерного образования — 2014: Сборник научных трудов международной научно-методической конференции (Октябрьский, 13 декабря 2014 г.). Челябинск: АРКАИМ, 2014.

7. Оленев А. А., Сагдеев К. М. Абстрактная и компьютерная алгебра. Ч. 1. Абстрактная алгебра в Maple: учебно-методическое пособие. Ставрополь: Бюро новостей, 2014.

8. Сагдеев К. М., Оленев А. А. Подход к математическому моделированию процесса выработки управляющего воздействия в организационных системах // Современные проблемы науки и образования. 2012. № 3.

9. Семакин И. Г., Шеина Т. Ю., Шестакова Л. В. Информатика и ИКТ. Профильный уровень: учебник для 10 класса. М.: БИНОМ. Лаборатория знаний, 2010.

Журнал «Информатика в школе»

Индексы подписки (агентство «Роспечать»)
на 1-е полугодие 2018 года

- 81407 — для индивидуальных подписчиков
- 81408 — для организаций

Периодичность выхода: 5 номеров в полугодие (в январе не выходит)

Редакционная стоимость:
индивидуальная подписка — 150 руб.
подписка для организаций — 300 руб.



Федеральное государственное унитарное предприятие "Почта России" Ф СП - 1													
Бланк заказа периодических изданий													
АБОНЕМЕНТ						На газету журнал							
Информатика в школе								(индекс издания)					
(наименование издания)								Количество комплектов					
На 20 18 год по месяцам													
1	2	3	4	5	6	7	8	9	10	11	12		
Куда													
				(почтовый индекс)				(адрес)					
Кому _____													
----- Линия отреза -----													
ПВ			место			литер			ДОСТАВОЧНАЯ				
									КАРТОЧКА			(индекс издания)	
На газету журнал						Информатика в школе							
						(наименование издания)							
Стои- мость		подписки		руб.		Количество комплектов							
		каталож- ная		руб.									
		переадре- совки		руб.									
На 20 18 год по месяцам													
1	2	3	4	5	6	7	8	9	10	11	12		
Город													
село													
почтовый индекс													
код улицы													
дом				корпус				квартира					
Фамилия И.О.													